A Review Paper on Object Detection Network using Deep- Reinforcement Machine Learning

Saurabh Tiwari[1], Dr. S. Veena Dhari[2]

[1]Research Scholar, [2]Associate Professor
Department of Computer Science and Engineering
RNTU Bhopal

***Abstract*: In this paper we introduce, and summarize many works and research papers on Reinforcement Learning and Deep Learning. Reinforcement learning and Deep Learning is an area of Machine Learning or Artificial Intelligence, it has an effective tool towards building artificially intelligent systems and solving d. Reinforcement learning was efficient in solving some control system problems. A particular machine learning method called deep learning has gained huge desirability, as it has obtained amazing results in many applications such as object detection, pattern detection, speech recognition, computer vision method, and natural language processing and other AI techniques. Much recent research has also been shown that deep learning methods can be club with reinforcement learning methods to study valuable representations for the problems with high dimensional raw data input.**

Our review paper focuses on the expansion of the Efficient Object Detection Network (EODNET). CNN model can provides fast object detection and recognition while saving resources like storage space, processing and memory. However, object detection techniques usually require either high power of processing or large accessibility of storage, making it tough for resource constrained plans to perform the detection in real-time without a connection to a server which have huge power. These margins allow for portable devices to attain high frame-rate object detection without the use of a Graphic Processing Unit (GPU). As an example of object detection application, presented in this paper shows the EODNET being used to detect and recognize the types of vehicle as an object.

***Keywords*: EODNET, Reinforcement Learning, Deep Learning, Artificial Intelligence, Machine learning.**

I. INTRODUCTION

Object detection is a broad research area in the recent time due to high demand of society. Reinforcement learning is a technique through interaction with an environment taking different actions and finding of many failures and successes while trying to achieve maximum rewards. Agent is not told which action to take. Reinforcement learning is similar to natural learning another branch of intelligence processes where a teacher or a supervisor is not available and learning process evolves with trial and error, different from supervised learning, in which an agent needs to know what the correct action is for every position it encounters and with same method it decide its action [1], [9].

Reinforcement learning (RL) algorithms involve the strategy of learning via interacting sequences of (actions, history, and rewards) with the environment. RL based methods have shown great successes in a variety of tasks from robotics [19] to resource allocation [43]. These have made them to be one of the main promising candidates to reach the goal of intelligence by which so many tasks can achieve in limited time, building those autonomous agents that can learn and produce unique results in complex and hesitant environments. This research includes brief to reinforcement learning which clear the intuition behind Reinforcement Learning in addition to the main model. After that, the outstanding power of Reinforcement learning is tinted. Accordingly, methods and the details for solving reinforcement learning problems are summarized.

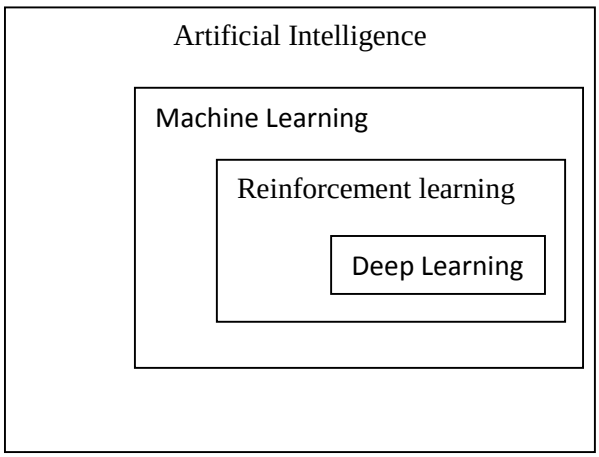

Fig 1. Artificial Intelligence Faces

Before an agent or robot (software or hardware) can select an action, it must have a good illustration of the environment in which the agent is to be learned [19]. Thus, perception is one of the key issue that must be solved before the agent can decide to select an optimal action to take and make move to other situation. Artificial Intelligence is associated with Deep Learning achievements in the current years. Deep Learning is basically a group of multiple layers of neural networks connected to one another. While Deep learning algorithms are the same as what was used in the late 1980's [4], deep learning progress is driven by the development of computational power and the tremendous increase of both generated and collected data [5]. Shifting from Central Processing Unit (CPU) to Graphics Processing Unit (GPU) [6] and later to TPU (Tensor Processing Unit) [7] accelerated processing speed and opened the door for more successes. However, computational capabilities are bounded by Moore's law [8] which may slow down building strong AI systems [9].

In recent years, research done in the deep learning area has shown that it is very promising and powerful tool to do automatic feature extraction from raw data. It has gained huge attraction not only in academic communities (because of its execution in various application like pattern recognition, speech recognition, computer vision, and natural language processing), but it has also applied in industry outcome by tech- giants like Google (Google's translator, Image search engine and many more), Apple (Apple's Siri), Microsoft (Bing voice search) and other big companies such as Facebook and IBM. Now days, Deep learning methods including supervised and unsupervised multilayer perceptrons (MLPs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) have started to use reinforcement learning methods. The evaluation of the resulting new algorithms and methods has indicated that deep learning methods can also be used to learn useful evaluation for reinforcement learning problems [29, 30, 31, 9, 27]. Consolidate Reinforcement Learning and Deep learning methods enables an RL agent to have a good realization of its environment by employing the possibilities that deep neural networks can provide.

The paper is organized as follows; Section II summarizes the Reinforcement learning Standard Model and Deep Learning. Section III summarizes the Associated Work Done. Section IV provides a detailed description of the EODNET, the training process and evaluates results. The conclusion and future work are shown in Section V (five).

II. REINFORCEMENT LEARNING STANDARD MODEL AND DEEP LEARNING

The main components of reinforcement learning model are policy, reward signal, value function and model [9], [10].

- The policy (π) is the way that the agent (something that perceives and acts in an environment [1]) will behave under certain circumstances. Simply the policy maps states into actions. It can be a lookup table, a function, or it may involve a search process. Finding the optimal policy is the core goal of reinforcement learning process [9], [10].
- The reward signal (R) shows how well and bad is an event and it defines the goal of the problem where the Agent intention is to maximize the total received reward. Accordingly, the reward is the key factor for updating the policy. Reward may be immediate or delayed, for delayed signals the agent need to determine which actions are more relevant to a delayed reward [9], [10].
- The value function is a prediction of the total future rewards, it is used to evaluate the states and select actions between accordingly [9], [10]The state-value function V(s) is the expected return when starting from a state s [9], [10]$V(s) = E(G_t / S_t = S)$ (1)

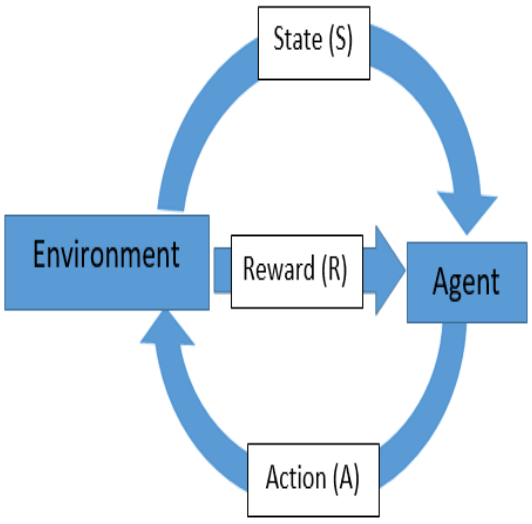

Fig.2 Reinforcement learning standard diagram

Where the return Gt is the total rewards R from time-step t. it is the sum of the immediate reward and discounted future reward [9], [10].

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$
$$= \sum_{m=0}^{\infty} \gamma^m R_{t+m+1} \qquad (2)$$

The discount γ represents the degradation factor of the future rewards when they are evaluated at present, γ ranges between 0 and 1. However, the use of discount is sometimes controversial.
The action value function q(s,a) is the expected return when starting from a state s and taking an action a. equation 3 shows the mathematical formulation [9], [10].

$$q(s,a) = \mathbb{E}(G_t \mid S_t = s, A_t = a) = \mathbb{E}\left(R_{t+1} + \sum_{m=1}^{\infty} \gamma^m R_{t+m+1} \mid S_t = s, A_t = a\right) \quad (3)$$

➢ The model of the environment allows predictions to be made about the behavior of the environment. However, Model is an optional element of reinforcement learning technique that use models and planning, are called model-based technique. On the other side are model free methods where the agent does not have a model for the environment. Model-free methods are explicitly trial-and error learners [9], [10].
Several works describes evaluating RL algorithms Duanet al (2016). Benchmark several RL algorithms and provide the community with baseline implementations. RL assessment metrics are proposed in (Whiteson et al 2011). Machado et al. (2017) revisit proposes Learning Environment to propose better evaluation methods in benchmarks. However, while the question of reproducibility and good experimental practice has been examined in related fields (Wagstaff 2012, Boulesteix, Lauer, and Eugster 2013, Stodden, Leisch, and Peng 2014, Bouckaert and Frank 2004, Bouckaert 2004, Vaughan and Wawerla 2012), to the best of our knowledge this is the first work to address this significant question in the context of deep RL.

Deep Reinforcement Learning: Recent breakthroughs in computer vision and speech recognition have relied on efficiently training deep neural networks on very large training sets. The most successful methods are trained orderly from the raw inputs, using lightweight updates based on stochastic gradient descent. By providing sufficient data into deep neural networks, it is often possible to learn better representations than hand crafted features [11]. These successes motivate our approach to Reinforcement learning (RL). Our goal is to find a reinforcement learning algorithm to a deep neural network which operates directly on RGB images and efficiently process training data by using stochastic gradient updates.

III. ASSOCIATED WORK

Currently, several works have shown the potential of CNNs in different areas. For example, some of them exploit these networks to create tools that help the visually impaired [35], [36]. The research presented in [35] uses neural networks to recognize products in markets. The Tensor- Flow library with a pre-trained Inception-v3 neural network was used to achieve high accuracy and fast training. In [36], Facebook shows a system that describes the content of images in audible text using deep CNNs to identify specie image details and convert them into words. The work seen in [37] uses neural networks to automatically identify tourist spots in Singapore photos. Images taken from the internet and trusty services were used for the training. In 2012, one of the most efficient CNN for image classification was proposed by Alex Krizhevsky. The model named Alex Net [29] consisted of five convolution layers, three fully-connected layers, 650,000 neurons and 60 million parameters. Using two CUDA GPUs, it provided one of the best results in the Image Net Large Scale Visual Recognition Competition (ILSVRC) [29]. R-CNN [38] is one of the most successful attempts to perform object detection within the image. Due to its high computational cost and slow detection, different approaches were proposed to improve the detection task. In [39], the results show that a hybrid region proposal method that uses the complementary information of color and edge detection can significantly improve the recall rate, boosting the accuracy of the entire system. Another hybrid approach, seen in [40] which combines feature extractors like SIFT with neural networks to reduce the training time. Alexe et al [41] presents a technique that, using sliding windows, perform an objectiveness measure on crops of the image to distinguish object windows from background ones. In [4], a segmentation strategy is used for object recognition, reducing the number of considered windows by 20 times. In [42] the sliding window method is used to evaluate regions with a CNN to count palm trees using unmanned aerial vehicle (UAV) imagery. A Direct evolution of R-CNN is the Fast R-CNN [43] that uses a region of interest (ROI) pooling layer to reduce the amount of forward passes per image from around 2000 to a single one by sharing the forward pass across its sub regions. Faster R-CNN [10] reduces the time taken by the object detection to less than half a second, while the accuracy is slightly increased over Fast R-CNN. It is done by replacing the selective search step by a region proposal network that takes as input the features that were already calculated by the forward pass, reducing the computational cost by sharing those features. In [44], different transfer learning techniques were compared with the complete training from scratch. The tested techniques were ne-tuning only the top layer, ne-tuning the top 3 layers, ne-tuning all the layers and training the whole network from scratch. For their dataset the best result was achieved by retraining only the top 3 layers. The usage of a CNN as feature extractor trained on a different dataset was explored on [45]. They used Alex Net [29] trained on ImageNet [17] and a recursive neural network with a softmax classier to classify the images. Some works make use of data augmentation techniques to improve training and avoid overt. In this process, several techniques are applied to generate images that will be used for training. In [31], a dataset of faces was used to synthesize new images with changes in pose, appearance and facial expressions. Different combinations of techniques were explored and presented significant gains compared to the original dataset. In [30] only the background and internal details of the image were changed. The original dataset was obtained using a green background so that the software could recognize it and make changes to create new images.

IV. PROPOSED EFFICIENT OBJECT DETECTION NETWORK (EODNET)

In our proposed method aim to achieve high speed detection or recognition that can participate in evaluation time with state-of-the-art networks, the EODNet the detection precision by using a shallower CNN model and by performing only a single pass through the image. Typically, networks for object detection uses over than 30 layers while EODNet only contains 09 layers. Rather than using different aspect ratio sliding windows for the detection, the proposed architecture assumes a fixed size convolution mask over the feature maps to perform a faster detection. These following steps are used to perform the proposed EODNet techniques.

First Step: Fetch and preprocess the dataset

Fetch collection of captured images of automobile containing one or more types of cars for this project. These images had their content manually segmented and classified with the aid of the software LabelMe [46]. Before being used, the images reduce the computational cost of the preprocessing phase; The dataset was then split in two subsets, training and testing datasets, so that the testing images are guaranteed to be seen only during the test. To reduce the overt and increase the amount of samples in the dataset, different techniques were used to create synthetic data. Neural networks are very sensible to transformations in the image, which can make it harder to detect and recognize objects in conditions different than the ones it was trained for. The techniques used to create synthetic data in this work were: altering the background of images by adding random noise, rotating objects, scaling the image, altering the aspect ratio, cropping random parts, transforming the internal texture, changing the brightness and the contrast. In general, training the network with a dataset that already contains those transformations allows the network to better generalize the features of each class and recognize them under those conditions. Using these techniques 25,000 images were obtained to train and test the network.

Second Step: **The execution for a single image**

First, the images that were resized to m X n during the dataset preparation are resized to A X B to further reduce the computational cost. This is the dimension that is used as input vector in the training and testing stage.
Since each pixel is a neuron in the first layer, reducing the size of the image also reduces the number of neurons and convolutions. Then, the image is processed by the EODNET that outputs class and confidence values for each detected region. Those regions are resized to match the original image size and bounding boxes are generated. Since the ratio and size of the feature map is different than the original image, the coordinates of the classified regions cannot be used directly in the original image. However, it is possible to obtain the ratio and coordinates in the original image by using cross-multiplication. if they overlap each other by 10% or more. To reduce the over-fitting, each fully-connected layer implements a dropout technique, which consists in temporarily removing some randomly selected neurons and all their connections during a forward pass to reduce co-adaptations on the training data [47]. The dropout greatly improves the performance of neural networks on a wide variety of applications, with the only downside being the increased training time [48]. After completing the detection phase, the regions are analyzed to remove low confidence detections. High confidence regions overlapping each other, if belonging to the same class, are merged to create larger bounding boxes. After completing, a new image is exported with bounding boxes highlighting the detected objects.

Third Step: Training and Testing dataset

The images are preprocessed by a script that prepares the dataset for training. The images are divided into regions that are classified either as part of an object or background. A CSV le containing the map of objects in each region is created so that the network can be trained to classify them. The model was implemented in TensorFlow [24] using Python for easy deployment on devices with different architectures.
To train a CNN from scratch, a significant amount of data with high variability is required. It increases the likely hood of the network to learn how to detect generalized low level features properly, increasing the accuracy on new data and reducing the overt. The intermediate layers of a CNN are usually feature extractors that can be generalized to different datasets and a common solution for the training is to use a bigger dataset to pre-train the network and then re-train the classification layers for the classes in the project [49]. This technique is widely used to achieve high accuracy when training networks on a limited dataset or limited amount of time. The dataset Dogs vs. Cats, from Kaggle [50], was used to pre-train the EODNet. It contains 25,000 images labeled either as dog or cat. The amount of data and the variability in colors and shapes in this dataset helps the generalization of the convolution filters.
A deep model like AlexNet [29] can take up to 11 GB of GPU memory, making it impossible to train from scratch in most of the GPUs. Since it is expected for EODNet to be used in resource constrained devices, the training was divided in three training phases. Moreover, the input images are reduced to 500000, for both training and testing, in order to reduce the amount of GPU memory required. The downsizing did not impair the object recognition in the case study. Each training phase takes part of the network with an extra fully connected layer (for output) and trains it for Kaggle's dataset [50], The first training phase takes the first four layers. For the second training phase, the weights trained on the first training phase are frozen while the rest of the convolution part is added to the network as trainable layers. After training the second training phase, the fully connected layers are added and the network is ready to be re-trained for detection using the target dataset. Allied to the batch size of 32 samples it was possible to train the network using less than 3 GB of GPU memory. These characteristics allow for the EODNet to be retrained by the user anytime, on regular desktop computers or laptops. To achieve faster convergence and avoid overt due to the order of the samples in the batches, the training data is shuffled for each period [51], [52].

Fourth Step: Result Calculation of dataset

The accuracy is used to measure how good the detections were. It is described by the following Eq:

$$\text{ClassificationAccuracy} = \frac{\text{true positive+true negative}}{\text{positive+negative}}$$

The calculations of the final results are made by comparing the bounding boxes proposed by the EODNet with the manually created ones (ground truth). To measure the exactness of object detection, the intersection over union (IoU) is used. It can be described as the Eq:

$$\text{IoU(A,B)} = \frac{|A \cap B|}{|A \cup B|} \quad \text{with } 0 \leq \text{IoU(A,B)} \leq 1$$

Where A and B are the bounding boxes manually found and detected by the CNN, respectively. The IoU can accurately describe the pixel-wise region similarity. It's an important metric for the tests on the network since the positioning and size of the bounding boxes are considered to be the weakest point of a model that uses a single pass for object detection. Detected objects are considered a true positive if they match the correct class and their bounding boxes overlap by at least 50%. The average IoU for a given class C, named IoUc, can be obtained as shown in the following Eq.:

$$\text{IoU}_c(C) = \frac{\sum_{bbox=0}^{\text{bboxes-1}} \text{IoU(bbox,ground truth)}}{\text{bboxes}}$$

With bboxes being the detected bounding boxes for all images in the dataset restricted to a given class. To better react the advantages of EODNet, a metric that takes into account the execution time is necessary. It can be described by the following Eq:

$$\text{Score} = \frac{\text{IoU} \times \text{Accuracy}}{\text{Execution Time}}$$

In order to execute the EODNet in a constrained device, all the training phases were performed on a laptop with Intel Core i7-HQ GHz CPU, GeForce GTX 970m, 4 GB, 500 GB drive, and windows 7. The tests were executed on a laptop with Intel Core i7- HQ GHz CPU, 500 GB 5400rpm hard disk, and windows 7.

Fifth Step: Results Analysis

Training the network from scratch on the automobile dataset, without the pre-training step, yielded inconsistent results and did not converge properly for most of the classes, achieving low accuracy on the cross-validation. The amount of original samples and the variability were not enough to train the whole neural network, what makes the pre-training a necessary step. Thus, the results from the experiments in this section were obtained on pre-trained networks. Thanks to the pre-trained model, it was possible to retrain the model for the food dataset in about 4 hours. This retraining resulted in the same accuracy obtained by the Faster R-CNN [38], with both being able to correctly classify all objects in the dataset, outputting bounding boxes.

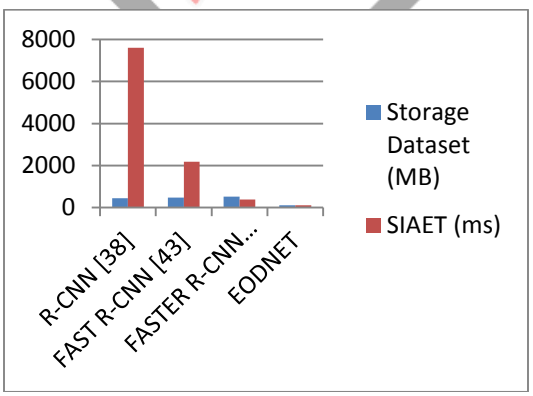

Fig 3. Result Analysis with EODnet

Prior to the version using a convolution layer over the feature map generated by the A X B convolution to perform the detection, the EODNet used a fixed sliding window over the input image. Although it had some advantages, such as the ability to work on different sized images, it greatly increased the evaluation time for the detection phase. The convolution layer over the feature map allows for better optimizations and since the feature map is much smaller than the image, the classier is used on a smaller batch of crops, resulting in a decrease of 98% on the single image average evaluation time (SIAET), as seen in fig 3. All the tests reported in this table were executed using the same dataset. The original R-CNN [38] uses SVM classifiers to recognize the objects after

being filtered by the network layers. Thus, the usage of storage space after training depends on the number of classes involved in the training set, often reaching over 2GB. Fast R-CNN [43] and Faster R-CNN [10], in the other hand uses a softmax classier which generates a le that is dataset independent. In comparison to those networks, the EODNet is shallower and uses less parameter. As seen in Graph, EODNet is also dataset independent in regards to the file size, which is about 5.5 times smaller. Since the image is only evaluated once and the detection regions have a fixed aspect ratio and size, the average time taken for a single image is 3.5 times lower than using Faster R-CNN [10].

V. CONCLUSIONS AND FUTURE WORK

Deep learning models with great power of automatically extracting difficult data representations from high-dimensional input data could outperform other state of the art of traditional machine learning methods. A major challenge in reinforcement learning is to learn optimal control policies in problems with raw visual input. Hierarchical feature extraction and learning distant representations of deep architectures, not only made the deep learning become a priceless tool for classification, but it has made it to be a valuable solution for the stated challenge in Reinforcement Learning tasks as well. Despite of the significant works done to data in combining Reinforcement Learning and Deep Learning, research on deep reinforcement learning is at its first ladder and there are still many uncharted aspects of this combination. Also, their challenges in real application such as robotics are yet unsolved and need more exploration to be done. More research is required on evaluate deep architectures both for end to end leaning, which performs a direct access to learn non-linear control policies, and deep state representation, which does dimension modification to present low dimensional representations then try to approximate Q-values. Especially, developing those mechanisms which make the end to end learning can be practical in real world application, those which doing a large number of actions is impossible.
Proposed EODNet is capable of quickly identifying the objects present in the image by evaluating each region of the image just once. Since it only requires a single pass throughout the image, the proposed network reduces the computational cost of the evaluation, which reduces the hardware requirement, energy consumption and execution time. Some trade-offs on the proposed architecture are expected, since it reduces resources usage. A future work can improve the positioning without impacting too much the performance of the detection. It would make the architecture suitable for even more applications, since most of them can make use of a faster detection.

REFERENCES

[1] S. Russell and P. Norvig, Artificial Intelligence A Modern Approach, 3rd ed. Pearson, 2009.
[2]A. M. Turing Computing machinery and intelligence, in Parsing the Turing Test Philosophical and Methodological Issues in the Quest for the Thinking Computer, 1950, pp. 23–65.
[3] S. M. Shieber, The Turing Test: Verbal Behavior as the Hallmark of Intelligence. Mit Press, 2004.
[4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533–536, 1986.
[5] A Goodfellow, Ian,Bengio, Yoshua, Courville, Deep Learning, 1st ed. Cambridge, MA Mit Press,2016.
[6] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "GPU Computing," Proc. IEEE, vol. 96, pp. 879–899, 2008.
[7] K. Sato, C. Young, and D. Patterson, "An in-depth look at Google's first Tensor Processing Unit (TPU)." [Online].Available:https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor processing - unit-tpu. [Accessed: 25-Dec-2017].
[8] R. R. Schaller, "Moore's law: past, present and future," IEEE Spectr., vol. 34, no. 6, pp. 52–59, 1997.
[9] R. S. Sutton and A. G. Barto, Reinforcement learning: an introduction, 2nd ed. Cambridge, MA: Mit Press, 2017.
[10] S. Ren, K. He, R. Girshick, and J. Sun, ``Faster R-CNN: Towards real-time object detection with region proposal networks," in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91-99.
[11] S. Scott and S. Matwin, ``Text classication using WordNet hypernyms," in Proc. Conf. Use WordNet Natural Lang. Process. Syst., 1998, pp. 38-44.
[12] D. G. Lowe, ``Object recognition from local scale-invariant features," in Proc. IEEE Int. Conf. Comput. Vis vol. 2. Sep. 1999, pp. 1150-1157.
[13] D. Anguita, A. Boni, and S. Ridella, ``Learning algorithm for nonlinearsupport vector machines suited for digital VLSI,'' Electron. Lett., vol. 35, no. 16, pp. 1349-1350, Aug. 1999.
[14] S. Karnouskos and A. W. Colombo, ``Architecting the next generation of service-based SCADA/DCS system of systems," in Proc. 37th Annu. Conf. IEEE Ind. Electron. Soc. (IECON), Nov. 2011, pp. 359-364.
[15] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, and K. Skadron, ``A performance study of general-purpose applications on graphics processors using cuda," J. Parallel Distrib. Comput., vol. 68, no. 10, pp. 1370-1380, 2008.
[16] S. Chetlur et al. (2014) cuDNN Eficient primitives for deep learning.**"**[Online].Available: https://arxiv.org/abs/1410.0759.
[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ``ImageNet: A large-scale hierarchical image database," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2009, pp. 248-255.
[18]R. Bekkerman, M. Bilenko, and J. Langford, Eds., Scaling up Machine Learning: Parallel and Distributed Approaches. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[19]Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-Based Learning Applied to Document Recognition. Piscataway, NJ, USA: IEEE Press, 2001, pp. 30-351.

[20]A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, ``CNN features off-the-shelf: An astounding baseline for recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, Jun. 2014, pp. 806813.
[21]D. E. Williams, G. E. Hinton, and R. J. Williams Learning representations by back-propagating errors Nature, vol. 323, no. 6088, pp. 533-538, 1986.
[22]K. Fukushima, ``A neural network model for selective attention in visual pattern recognition," Biol. Cybern., vol. 55, no. 1, pp. 5-15, 1986.
[23]X. W. Chen and X. Lin Big data deep learning Challenges and perspective IEEE Access, vol. 2, pp. 514-525, 2014.
[24]M. Abadi et al. (2016) TensorFlow Large-scale machine learning on heterogeneous distributed systems [Online] Available: https:// arxiv.org/abs/1603.04467.
[25] J. Bergstra et al., ``Theano: A CPU and GPU math compiler in python," in Proc. 9th Python Sci. Conf., 2010, pp. 1-7.
[26] R. Collobert, K. Kavukcuoglu, and C. Farabet, ``Torch7: A MATLAB-like environment for machine learning," in Proc. BigLearn, NIPS Workshop, 2011, paper EPFL-CONF-192376.
[27]P.F. Felzenszwalb, R. B. Girshick, and D. McAllester,"Cascade object detection with deformable part models," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2010, pp. 2241-2248.
[28] F. E. H. Tay and L. Cao, ``Application of support vector machines in nancial time series forecasting," Omega, vol. 29, no. 4, pp. 309-317, Aug. 2001.
[29]A. Krizhevsky, I. Sutskever, and G. E. Hinton,``ImageNet classication with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097-1105.
[30]B. Sapp, A. Saxena, and A.Y. Ng, ``Afast data collection and augmentation procedure for object recognition," in Proc. AAAI, Chicago, IL, USA, 2008, pp. 14021-408.
[31]I. Masi, A. T. Tran, J. T. Leksut, T. Hassner, and G. Medioni. (2016). ``Do we really need to collect millions of faces for effective face recognition?" [Online]. Available: https://arxiv.org/abs/1603.07057
[32]Z. Zhang, J. Warrell, and P. H. S. Torr, ``Proposal generation for object detection using cascaded ranking SVMs," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2011, pp. 1497-1504.
[33]J. Leng, T. Li, G. Bai, Q. Dong, and H. Dong, ``Cube-CNN-SVM: A novel hyperspectral image classication method," in Proc. IEEE 28th Int. Conf. Tools Artif. Intell. (ICTAI), Nov. 2016, pp. 1027-1034.
[34]S. Albers, ``Energy-efcient algorithms," Commun. ACM, vol. 53, no. 5, pp. 86-96, 2010.
[35]S. C. Hoffman and D. Thiagarajan, ``Continuity report: Revisiting grocery recognition using tensorow," to be published.
[36]D. G. García, M. Paluri, and S. Wu. Under the Hood: Building Accessibility Tools for the Visually Impaired on Facebook. [Online]. Available: https://code.facebook.com/posts/457605107772545/under-thehood- building-accessibility-tools-for-the-visually-impaired-on-facebook/.
[37]L. F. D'Haro, R. E. Banchs, C. K. Leong, L. G. M. Daven, and N. T. Yuan, ``ALEXIS: Automatic labelling and metadata extraction of information for Singapore's images," to be published.
[38]R. Girshick, J. Donahue, T. Darrell, and J. Malik, ``Rich feature hierarchies for accurate object detection and semantic segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580-587.
[39]R. Qian, Q. Liu, Y. Yue, F. Coenen, and B. Zhang, ``Road surface trafficsign detection with hybrid region proposal and fast R-CNN," in Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD), 2016, pp. 555-559.
[40]M. Merler, C. Galleguillos, and S. Belongie, ``Recognizing groceries in situ using in vitro training data," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2007, pp. 1-8.
[41]B. Alexe, T. Deselaers, and V. Ferrari, ``Measuring the objectness of image windows," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 11, pp. 2189-2202, Nov. 2012.
[42]E. K. Cheang, T. K. Cheang, and Y. H. Tay. (2017). ``Using convolutional neural networks to count palm trees in satellite images." [Online]. Available: https://arxiv.org/abs/1701.06462
[43]R. Girshick, ``Fast R-CNN," in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 1440-1448.
[44]A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, ``Large-scale video classication with convolutional neural networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1725-1732.
[45]H. M. Bui, M. Lech, E. Cheng, K. Neville, and I. S. Burnett, ``Object recognition using deep convolutional features transformed by a recursive network structure," IEEE Access, vol. 4, pp. 10059-10066, 2016.
[46]B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, ``LabelMe: A database andWeb-based tool for image annotation," Int. J. Comput. Vis., vol. 77, nos. 1-3, pp. 157-173, 2008.