

# Adaboost Cascade Classifier for Classification and Identification of Wild Animals using Movidius Neural Compute Stick

S.Divya Meena, L.Agilandeeswari



**Abstract:** Deep learning has gone deeper and has been utilized in almost all the applications like object recognition, image classification, speech recognition and much more. Most of the real-time applications rely on deep learning for accurate results. But one downside to the deep learning is its demand for GPUs (Graphical Processing Unit) or TPUs (Tensor Processing Units) for faster execution. There was no one-stop-shop hardware and software for deep learning applications, until recently Intel launched the Movidius Neural Compute Stick (NCS). This sleek device provides the power of GPU in a CPU based system. In this work, we have modeled an animal detection system using NCS and AdaBoost classifier powered by Multi-Block Local Binary Pattern (MB-LBP) features. The model has been built upon AlexNet and has achieved an average accuracy of 96.8% and a false rate 2.3% in classifying the animals as wild and non-wild. Furthermore, the model has a speedup of 467% when compared to the execution in the CPU based system.

**Keywords:** Deep Neural Network, Alex Net, Movidius Neural Compute Stick, MP-LBP, AdaBoost Classifier.

## I. INTRODUCTION

Image classification, one of the popular computer vision problem, classifies the given object or the instances in the image into one of the predefined classes or categories. Conversely, object detection can identify multiple instances in the same image, i.e., it localizes the objects in the image. Traditional approaches in object detection rely on complex algorithms that use features like edges, texture, curves, etc. [1-4] to recognize the objects. Contemporary techniques like Deep learning uses a rather interesting approach in which there are multiple layers and the neurons in each layer are interconnected, thereby forming complex layers [5]. Each layer of the architecture focuses on capturing different features and they use various algorithms specific to their feature descriptors. For instance, to detect a traffic stop sign, we would use a deep learning architecture, in which one layer will detect the edges, another layer will detect the color, yet another layer will detect the corners and so on [6]. This capability of the Deep Neural Network to split the given task

into multiple layers of the modest algorithm makes it more efficient [7] for solving real-time applications.

The main objective of this work is to detect the various animals from the given image. The model is intended to detect various kinds of animals like deer, monkey, pig, fox, etc. The model has been widely tested on both RGB and thermal images and has produced promising results that could be applied for real-time animal detection applications. The model is built using Intel® Movidius™ Neural Compute Stick (NCS) [8], a miniscule deep learning based USB drive. NCS is driven by Movidius Visual Processing Unit (VPU), a high performance device that requires very low power. NCS is capable of translating a PC environment based trained neural networks to various apps [9-11]. Indeed, the VPU is already available in millions of products like security cameras, drones, industrial machine vision equipment, etc. [12-14]. The Movidius NCS can run networks that are based on Caffe and Tensorflow. NCS optimizes the networks to be able to run in the ultralow power Myriad™ 2 VPU. Generally speaking, Movidius NCS is an easy-to-use kit that could be thought of like a Graphical Processing Unit (GPU) running in a USB. The notable feature of NCS is that it supports Convolutional Neural Networks (CNN) profiling, modeling and fine-tuning.

While Neural Compute Application Zoo (NC App Zoo) has a number of pre-trained DNN models, it is always not a good choice to completely adopt a pre-trained network. Instead, we can build a custom model that is more appropriate for our application. NC App Zoo has several pre-trained models such as MobileNets, SqueezeNet, AlexNet, GoogleNet, etc [15-17]. These networks are mostly trained on ImageNet dataset and are quite useful in building modest projects. But, for building a Proof of Concept (PoC) application, we need to customize the neural network.

In this work, we will train a pre-defined neural network for our custom dataset, profile the model using Movidius NCS Software Development Kit (SDK), and amend the network for the least execution time and finally deploy the customised neural network. Our model is a fine-tuned version of AlexNet. The fine-tuned model is trained on ImageNet dataset. Movidius VPU is used for the inference.

## II. RELATED WORK

While automatic animal identification is a common task, localizing the animal with a bounding box is very rare among the existing relevant literature. The few existing animal detection systems developed are also trained on very images and tested for very few common classes of animals.

Manuscript published on 30 December 2019.

\* Correspondence Author (s)

S.Divya Meen\*, Research Scholar, School of Information Technology and Engineering, VIT University, Vellore, India. Email: [sdivya.meena2017@vitstudent.ac.in](mailto:sdivya.meena2017@vitstudent.ac.in)

Dr.L.Agilandeeswari, Associate Professor, School of Information Technology and Engineering, VIT University, Vellore, India. Email: [agila.l@vit.ac.in](mailto:agila.l@vit.ac.in)

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

# Adaboost Cascade Classifier for Classification and Identification of Wild Animals using Movidius Neural Compute Stick

Moloti et al. developed GroceryNet [26], a fine-tuned CaffeNet model for detecting groceries using Intel Movidius Neural Stick. Nashwan et al. also used Movidius NCS for object detection and recognition application [27]. Banumathi et al. developed a real-time vehicle detection application [28] using Movidius NCS. Recently, Sergio et al. analysed the possibilities of Vision Processing Unit as Co-processor for inference and concluded that multi-VPU is equivalent to GPU [29].

Norouzzadeh [18] used deep neural networks on the Snapshot Serengeti dataset to train the images. They achieved an accuracy of about 92% in classifying 48 species of animals from Tanzania's Serengeti National Park. The accuracy rate for the same dataset was further improved by Alexander Gomez et al. Alexander Gomez et al [19] also utilised very deep convolutional neural network for classifying 20 different animal species from Snapshot Serengeti dataset. Using residual network (ResNet) topology, they achieved an accuracy of about 88.9% in Top-1 category and about 98.1% in Top-5 category. In identifying animals from Camera-trap images, many [20] has achieved a decent accuracy. But, their models were based on manually designed feature extraction and moreover they worked only with few thousands of images. Though feature extraction is one of the preliminary steps in image processing, their use in automated image classification is limited and hence the works suffered from low accuracy rates.

Yu et al [21] has manually cropped the animal in the image and selected those images that contain the whole image of an animal for classification. With this, they were able to achieve an accuracy of about 82% in classifying 18 species of animals. But this technique was not suitable for automatic classification, as it relied on manual cropping. In contradiction, Chen et al [22] utilised convolutional Deep Neural Network to automate the process of segmentation and identification. They were able to achieve accuracy of about 38.3% in classifying 20 classes of animals. This accuracy is far from the desired rate.

Wawerla et al [23] detected wild bears from videos captured at the Arctic Circle. They developed a novel algorithm called motion shapelet, which is nothing but the extension of shapelet algorithm developed for pedestrian detection. The model was trained on 405 positive images and achieved an accuracy of 76%. Zhang et al [24] detected animals from highly cluttered images using a joint technique of Deep CNN and HOG features. The model was trained on 800 camera-trap images and has achieved an accuracy of 82%.

Parham et al [25] detected zebra using the state-of-the-art techniques like faster r-CNN and YOLO v1. They have addressed the challenging conditions in camera-trap images. The dataset of 2500 images was manually annotated and was trained on around 2672 images. The model has achieved an accuracy of 55.6% and 56.5% with faster r-CNN and YOLO v1 respectively.

## III. METHODOLOGY

In this section, we will discuss the steps involved in deploying an animal detection model in NCS. The model follows the basic four steps of training, profiling, fine-tuning and running inference. In addition, the model can be deployed in any edge devices too. Furthermore, we classify the Area of Interest with AdaBoost classifier in combination with MB-LBP feature extraction.

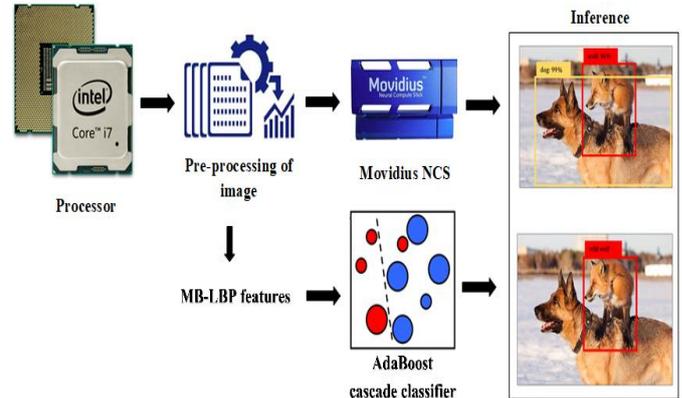


Fig. 1. Proposed animal detection model

### A. Animal detection with Movidius NCS

NCS has several pre-trained model in its official repository, but in order to develop our own model, one has to initially develop the architecture in CPU based system and train the dataset. The model's inference can then be obtained by shipping the model from CPU to NCS. The test images has to be pre-processed in the CPU system itself and only one image can be loaded at any point of time. However, the inference is so quick just like a GPU based system. Fig 1 depicts the flow of animal detection model. The images are initially loaded from the CPU based system and pre-processed before offloading them to the Movidius NCS. We have utilized the existing AlexNet model from NCS and hence the model need not be ported, the inference is finally obtained for the given test image.

#### 1. Train

Fine-tuning a pre-trained neural network, acclimates the architecture and trains the model from the learned weights. We use AlexNet as our base network and since ImageNet has 1,000,000 images, we use its pre-trained weights into our model. We modify the last layer of the network according to the number of classes in our model. Additionally, we decrease the learning rate for all the layers except for the newly introduced layers. This way, the new layers can learn fast and all other layers will change slowly. By deriving the parameters of AlexNet, our model was able to converge fast than expected. Also to avoid the problem of over-fitting, we monitored the trend line of training loss. For a better inference, we reviewed the detection accuracy to around 55,000 iterations. The model was able to achieve 96.8% accuracy in animal detection. Dataset preparation is one of the important steps in training. We follow the basic pre-processing steps like resizing of image, cropping to get the area of interest, histogram equalization, etc. AlexNet uses an image dimension of 227x227 pixels. So, we resize all the images to 227x227 for training. We use mean subtraction technique to normalize the data. We used a mean value of Red=123, Green=117 and Blue=102 for the RGB data.

#### 2. Profile

The standard NCS SDK contains a profiler tool called mvNCPProfile, to analyse our neural network. This tool gives us the layer-by-layer explanation of our trained network. It gives a detailed explanation of the bandwidth requirement, execution time and the complexity of each layer. The weight file of our network is utilized for this purpose. Fig 2 depicts the network profile.

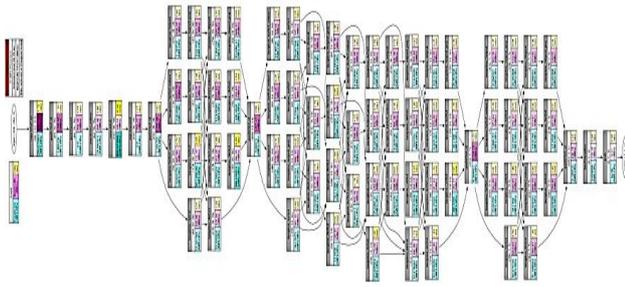


Fig. 2. Network profile

### 3. Fine-tuning and running inference

The results of the profiler can be modified to get best results in least execution time. The fine-tuned model can then be used to get the inference. Initially, we load the test image from the Ubuntu folder and unload it to NCS, which acts as a visual co-processor just like the VPU. The processing of neural network is done exclusively by the NCS, thus emancipating the application's processor and memory for other low-level tasks. One important task to be taken care of is the pre-processing of image before loading the image onto NCS. The inference results can be read either using blocking or non-blocking function call. By default, we use blocking call to read the inference results.

### B. Animal classification with AdaBoost classifier

The animals that are detected by Movidius NCS is further classified using AdaBoost classifier. We utilize the AdaBoost-based training algorithm to train a cascade classifier that classifies the animal as either wild or non-wild.

### 1.Feature extraction with MB-LBP

Multi-Block Local Binary Pattern is utilized for extracting the features. Typically, it works by comparing the average pixel value with multiple pixels from various blocks. The averaging function removes the noise and this function is also available in Multi-scale Local Binary Pattern. Feature types such as Multi-scale Block LTP also uses the averaging function for removing the noise from the inference. The Local Binary Pattern is calculated from all the image patches in every possible position. This will produce a response histogram. This function is really time-consuming and is not the wise choice for real-time detections. Moreover, the feature types from LBP cannot be directly used by cascade classifier, since cascade classifier uses histogram representation. To overcome these challenges, Zhang et al proposed Multi-Block Local Binary Pattern, which is compatible for cascade classifiers like AdaBoost and also it has faster detection rate. In MB-LBP, we calculate the average pixel value from the central multiple blocks and this average value is compared with the surrounding pixel values for getting the response pattern. The response pattern from MB-LBP can be directly used by the cascade classifier and the histogram representation is not required.

### 2. AdaBoost classifier for AoI classification

The animals that are detected by Movidius NCS is further classified using AdaBoost classifier. We utilize the AdaBoost

training algorithm for training the cascade classifier which will then classify the animal either as wild or non-wild. The input will include the sample and feature pool and the output will be a cascade classifier with many strong classifiers. The positive and negative patches of both thermal and visible images are taken as positive and negative samples respectively. The features extracted from MB-LBP will be in feature pool. The algorithm builds one strong classifier by selecting the best features repeatedly from the feature pool. With these features, we build a weak classifier for this particular strong classifier. The construction of one strong classifier will be done only when it meets the requirement of the predefined minimum detection rate and the maximum false alarm rate. Hence, the strong classifier will contain many feature types.

## IV. RESULTS AND DISCUSSION

In this section, we discuss about the results and also compare the model with various other architectures. The comparative results are also presented in this section. Though the model was trained on ImageNet dataset, we used our own images for testing. The images are manually captured by us using FLIR e40 thermal imaging camera. The captured images are in thermal form but it can be converted to visible image form also. For this work, we utilized the visible image form of animals. The inference from the animal detection model is depicted in Fig 3. The test images are taken with FLIR camera, where we can read the temperature of each and every pixels in the image. However, the temperature can be read only in the thermal form of the image. The results from the model detects the animal and draws a bounding box around it and displays the class of animal with its confidence score.

Object detection with NCS has significant improvement in speedup when compared to CPU solely. Furthermore, AlexNet has faster execution time when compared to GoogleNet and Inception network, making it the best model for accurate and real-time animal detection. Table 1 discusses the performance of various models like AlexNet, GoogleNet and Inception network. The models execute faster with NCS than CPU alone and the speedup is relatively high in AlexNet than the counter models. Animal detection with NCS is obviously faster than CPU alone. Besides, AlexNet performs way better than GoogleNet and Inception model. It is to be noted that, GoogleNet takes around 0.60 sec for detecting even with NCS; whereas AlexNet takes only 0.51 sec even without NCS. This shows that AlexNet is much better in capturing the detailed characteristics of animals than other two models. Also, Inception model is currently not supported by NCS. The average precision of various animal classes is depicted in Fig 4.

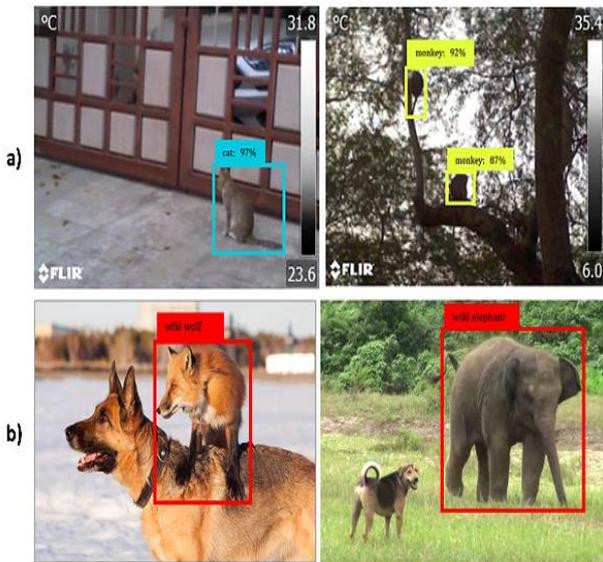


Fig. 3. Inference from animal detection model a) Movidius NCS on FLIR images b) AdaBoost classifier

Table- I: Performance analysis of various models via CPU and NCS

Parameters	AlexNet		GoogLeNet		Inception	
	PC	PC+NCS	PC	PC+NCS	PC	PC+NCS
Average time/image	0.51	0.37	0.85	0.60	0.68	Model not available in NCS
CPU usage	83%	31%	96%	57%	71%	
RAM usage	~3.47 GB	~1.52 MB	~3.76 GB	~1.98 MB	~3.60 GB	

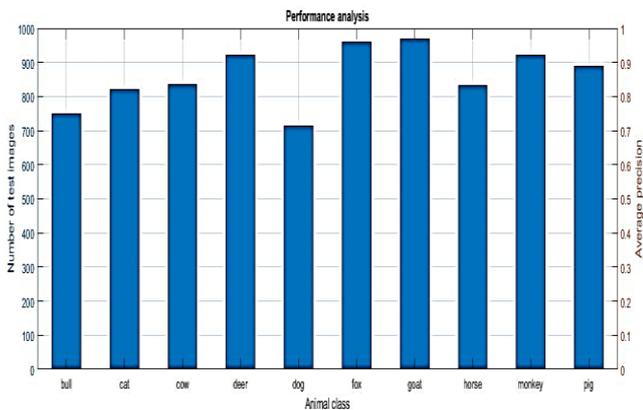


Fig. 4. Average precision of various classes of animals

The model is trained for 10 classes of animals like cat, dog, cow, bull, deer, fox, goat, horse, monkey and pig. The above figure depicts the average precision of various classes of animals along with the number of test images for each class of animal. Left side of the figure denotes the number of test image. For instance, bull has 750 images approximately. Similarly, right side denotes the average precision. Goat class had the highest precision rate of about 98% and dogs had the least precision of 0.79. This is because, the test images of dogs was specifically chosen to include various breeds of dogs. The model was trained for few breeds of dogs like German shepherd and Pomeranian but was tested with many different dog breeds like pug, hound dog, wolf dog, Canidae, etc. The test images of dogs had higher intra-class variance with fox and cat and this caused the model to have lower precision for dog class. However, the average accuracy of the

model was 96.8%, thus making it the most appropriate model for real-time animal detection system.

V. CONCLUSION

Object detection is one of the interesting problems in computer vision due to the various challenges in the image. Recently, Deep Neural Network has started evolving as one of the best solution for object detection. DNN has been the back-bone for any real-time object detection applications. In particular, animal detection is one of the trickiest problem because of the challenges under which the animals are captured. The very nature of animals is one of the biggest challenges in detecting them. Despite the challenges, DNN promises to produce accurate detection even under few challenging conditions like lighting effect, occlusion, cluttered background, etc. Despite being accurate in detecting, the model should also be fast in detecting. Intel Movidius NCS satisfies this requirement. With NCS, we were able to achieve a speedup of around 467% when compared to mere PC based model.

The choice of network plays a major role in the model. Once, an appropriate base network is chosen, we can then customize them according to our requirement. CNN has a lot of hidden parameters and the main difference between various models is the choice of parameters. Dataset with well-defined annotations have better performance than deeper CNN but there exist a trade-off between the execution time and confidence score. In future, deeper CNN can be exploited to get better accurate detections.

REFERENCES

1. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding".
2. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS, 2012.
3. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going Deeper with Convolutions".
4. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition", arXiv preprint arXiv:1512.03385, 2015.
5. K. K. A.F. O'Connell, J.D. Nichols, Camera traps in animal ecology: Methods and analyses. Springer, Jan. 2011.
6. R. Kays, B. Kranstauber, P. A. Jansen, C. Carbone, M. J. Rowcliffe, T. Fountain, and S. Tilak, "Camera traps as sensor networks for monitoring animal communities," LCN, p. 811–818, Oct. 2009.
7. A.Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer, "Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna," Scientific data, vol. 2, June 2015.
8. <https://software.intel.com/en-us/movidius-ncs>
9. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems, pp. 91–99, Curran Associates, Inc., 2015.
10. Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on, 2014.
11. Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on, 2014.



12. Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. Advances in Neural Information Processing Systems 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 2553–2561, 2013.
13. Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. CoRR, abs/1312.6229, 2013.
14. Andrew G. Howard. Some improvements on deep convolutional neural network based image classification. CoRR, abs/1312.5402, 2013.
15. Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25, pages 1106–1114, 2012.
16. Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on, 2014.
17. Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V. Le, and Andrew Y. Ng. Large scale distributed deep networks. In P. Bartlett, F.c.n. Pereira, C.j.c. Burges, L. Bottou, and K.q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1232–1240. 2012.
18. Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M.S., Packer, C., & Clune, J. (2017). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. 1-17.
19. Alexander Gomez. Augusto Salazar. Francisco Vargas. (2017). Towards Automatic Wild Animal Monitoring: Identification of Animal Species in Camera-trap Images using Very Deep Convolutional Neural Networks. Ecological Informatics, 41(), 24-32. doi: <https://doi.org/10.1016/j.ecoinf.2017.07.004>
20. Swinnen, KRR. Reijniers, J. Breno, M. Leirs, H. (2014). A novel method to reduce time investment when processing videos from camera trap studies. PLOS ONE e98881. Doi: <https://doi.org/10.1371/journal.pone.0098881>.
21. Figueroa, K. Camarena-Ibarrola, A. García, J. Villela, HT. (2014). Fast automatic detection of wildlife in images from trap cameras in Iberoamerican Congress on Pattern Recognition. Springer, 940–947. Doi: 10.1007/978-3-319-12568-8\_114
22. Yu, X. Wang, J. Kays, R. Jansen, P.A. Wang, T. Huang, T. (2013). Automated identification of animal species in camera trap images. EURASIP Journal on Image and Video Processing, 2013(52), 1-10. Doi: <https://doi.org/10.1186/1687-5281-2013-52>
23. Wawerla, J., Marshall, S., Mori, G. et al. Machine Vision and Applications (2009) 20: 303. <https://doi.org/10.1007/s00138-008-0128-0>
24. Zhao, Z., Zheng, P., Xu, S., & Wu, X. (2018). Object Detection with Deep Learning: A Review. IEEE transactions on neural networks and learning systems.
25. Parham, J.R., & Stewart, C.V. (2016). Detecting plains and Grevy's Zebras in the realworld. 2016 IEEE Winter Applications of Computer Vision Workshops (WACVW), 1-9.
26. Moloti Nakampe (2018), "GroceryNet at the Edge using Intel Movidius Neural Compute Stick", DOI: 10.13140/RG.2.2.13745.43360.
27. N. A. OTHMAN and I. AYDIN, "A New Deep Learning Application Based on Movidius NCS for Embedded Object Detection and Recognition," 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, 2018, pp. 1-5.
28. K. Banumathi, B. Bharathi, S. Daisy Deve Priya, P. Gogulalakshmi, (2018), "Real Time Vehicle Detection using Movidius Neural Compute Stick", International Journal of Engineering Research & Technology (IJERT), RTICCT - 2018 Conference Proceedings, Volume 6, Issue 08.
29. S. Rivas-Gomez, A. J. Pena, D. Moloney, E. Laure and S. Markidis, "Exploring the Vision Processing Unit as Co-Processor for Inference," 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Vancouver, BC, 2018, pp. 589-598.



SECOND AUTHOR, completed her PhD and working as HOD & Associate Professor in the Department of Digital Communications, School of Information Technology & Engineering (SITE), VIT Vellore. She also got best researcher award for the past four years from 2015 to till

date. She received her Bachelor's degree in Information Technology and Master's in Computer Science and Engineering from Anna University during 2005 and 2009 respectively.

Dr. Agilandeewari is having around 13+ years of teaching experience and published 50+ papers in the peer reviewed reputed journals. Her reputed publications includes research articles in peer reviewed journals namely Expert Systems with Applications, Journal of Ambient Intelligence and Humanized Computing, Multimedia Tools and Applications, Journal of Applied Remote Sensing. She is the peer reviewer in journals include IEEE Access, Array, Artificial Intelligence Review, Informatics in Medicine Unlocked, Neuro computing, Computers and Electrical Engineering, Journal of King Saud University – Computer and Information Sciences, IET Review, Journal of Engineering Science and Technology (JESTEC), etc. She also published about 13 engineering books as per Anna University Syllabus. She is a life time member in Computer Society of India. Her areas of interests include Image and video watermarking, Image and Video processing, Neural networks, Fuzzy Logic, Machine Learning, Cryptography, IoT, Information Centric Networks and Remote Sensing.

### AUTHORS PROFILE



FIRST AUTHOR, received the B.Tech degree in Information Technology from Vellore Institute of Technology, Vellore, India, in 2014 and the M.E. degree in Computer Science and Engineering from Anna University, Vellore, India, in 2016. She is currently