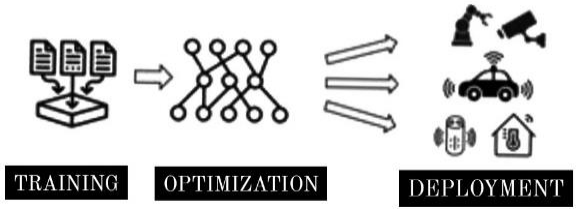
# Optimizing Neural Network for Deployment on Edge Devices

Nikhitha LP[1], Kunal Pandya[2], Satwik DP[3], Sajala MP[4], Indushree M[5]
[1,2,3,4] *Dept. of ISE, Global Academy of Technology, Bangalore*
[5]*Assistant Professor, Global Academy of Technology, Bangalore*

***Abstract*—Networks like convolution neural networks and their derivatives have contributed to the area of computer vision's explosive growth in recent years. Such a network however cannot be deployed on the edge device because of the high computational cost and memory requirements for model storage. Edge computing can address problems with latency, connectivity, cost, and privacy, but edge devices still face difficulties due to the deep learning model's high resource requirements. A big network-sized CNN model with more floating-point operations is required, particularly for deep learning-based applications. In order to address those issues, this study provides a strategy for deploying deep learning models to edge devices. The neural network has been optimized utilizing memory and computation-saving methods like pruning, weight clustering and quantization.**

## I. INTRODUCTION

The mapping of input to output using mathematical formulas is known as deep learning, which is a subset of machine learning. In order to create a relationship between the input and the output, these functions are able to extract non-redundant information or patterns. Computers can now complete activities that come naturally to people thanks to deep learning. It deals with algorithms whose development was affected by the structure and functionality of the human brain. Deep learning is a technique that enables a computer model to learn categorization tasks directly from images, text, or music. In order to train deep learning models, large datasets and neural network topologies that automatically extract features from data are used. The state of the art in several disciplines, including speech and vision, has drastically advanced because of deep learning. Because of its capacity to manage unstructured data and spot its unexpected patterns, deep learning is flourishing. Most Deep Learning-based apps currently in use are cloud-based and utilize a powerful GPU. In the future, there will be billions of connected devices that will continuously serve the business and our daily life. Internet-of-Things (IoT) and connectivity technologies like 5G, which are both rapidly developing, have made this conceivable. In order to fully realize the promise of edge big data, these devices will simultaneously produce a significant amount of vital data points that are supplied at the network edge. This will require both quick data processing and cognitive data analysis. Traditional cloud computing and on-device computing are unable to solve this issue due to a lack of processing power and latency. It is true that the problem is being clarified by the growing use of edge computing, which moves data processing from a remote network core to a local network edge, considerably reduces latency and thereby improves and increase the efficiency.




## II.LITERATURE REVIEW

### *A. Bird Species Identification*

K.Sireesha et.al [13], Deep Learning method was used to solve the issue of not being able to identify the bird species using TensorFlow and deep convolutional neural network algorithm. Using the tensor flow and the deep convolutional neural network algorithm, an image is converted into a greyscale format and many nodes of comparison are generated. These various nodes are compared to the testing dataset, and a score sheet is produced. The score sheet will then help predict the species of the bird using the highest score. TensorFlow creates a graph that consists of processing nodes that perform certain operations such as mathematical operations and represent links to other

nodes. This graph makes performing complex operations more easy.

*B. Covid-19 Disease Detection .*

Shayan Hassantabar et.al [5], X-ray pictures of the lungs was used to identify and diagnose COVID-19 patients using three deep learning-based approaches. Two algorithms were proposed for the diagnosis of the disease, deep neural networks on the features of images and convolutional neural networks techniques using lung images. Two AI-based diagnostic techniques were applied. Features were extracted and classified using ANN, and the efficiency and sensitivity of the method were examined using CNN. In order to assess the classification, accuracy, specificity, recall or sensitivity (R), S-dice and miss-rate, fall-out, and Jaccard similarity value are utilized. CT scan and chest x-ray data can be used to train and test a convolutional neural network. CNN have three main layers that make it up, convolution layer, pooling layer and dense layer that help it with the feature detection. This project makes use of three alternating convolution and pooling layers. Different DNN's such as Inception, MobileNet and ResNet were used to compare it to the model that was proposed. The proposed DNN out performed all the other DNN's.

*C. DNN For Face Mask Detection.*

Preeti Nagrath et.al [11], the Covid-19 epidemic's preface, face mask discovery has made considerable strides in the fields of image processing and computer vision. A multitude of strategies and methodologies have been used to develop several face discovery algorithms. Advanced literacy, the recommended fashion in this study to honor face masks uses TensorFlow and OpenCV. The Single Shot Multibox Sensor system(SSDMNV2) makes use of the MobilenetV2 armature as a foundation for the classifier and the Single Shot Multibox Sensor as a face sensor. This system may be utilised in bedded systems like the NVIDIA Jetson Nano and the jeer Pi to do real- time mask discovery because of how featherlight it is. The dataset offered in this study, which was gathered from numerous sources, can be used by other experimenters to make more complicated models for facial corner, face recognition, and element identification ways.

*D. Recognition of Gesticulation by using Tensorflow*

Zixian Zeng et.al [1], the study creates a gesture recognition model with a convolution network model based on Google's most current open-source Tensorflow framework. It also outlines the characteristics of the Tensorflow platform. The project aims to merge a self-collected dataset with a previously known dataset. Experiment results reveal that the model has a high degree of resilience, high computational efficiency, high recognition accuracy, and the ability to select the optimal model and improve gesture recognition.

*E. Interaction Speed of the Neural Network with Tensor flow*

Kristian Dokic et.al [2], TensorFlow that is one of the most well-liked machine learning frameworks, was initially created for microcontrollers by its creators. The number of neurons on a well-known microcontroller (Arduino Nano 33 BLE Sensor) and the speed of fully connected neural networks are investigated in this study, as is the effect of quantization on neural network weights. It was expected that the selected quantization would result in a four-fold decrease in the size of the model two hidden; however, this was only attainable with a large number of neurons in TensorFlow Lite.

*F. Application of EDGE computing*

Fangxin Wang et.al [3], the world of the future was envisaged as a connected reality where lots of bias will continuously serve the business and our diurnal lives. The burgeoning growth of the Internet of effects(IoT) and connectivity technologies like 5G has made this conceivable. This bias will all work together to induce a substantial to completely achieve the pledge of edge big data, it's necessary to do both rapid-fire data processing and cognitive data analysis due to the cornucopia of significant data points at the network edge. Traditional pall computing and on- device computing is both inadequate for resolving this issue due to their high quiescence and limited processing capability. The expanding use of edge computing, which shifts data processing from a away network core to a original network edge, significantly reduces quiescence and boosts effectiveness.

*G. Analyzing overfitting mitigation for leaf disease*

Serosh Karim Noon et.al [4], deep literacy models have revolutionized image processing during the once

20 times. These models have lately been effectively used to identify factory splint conditions. Deep literacy models are complex and prone to overfitting, however. To help overfitting, it's pivotal to choose the right training surroundings. In this study, we probe the impact of data addition and colorful optimization styles applied to a pretrained DenseNet- 121 model in terms of overfitting.

*H. Classifying ECG for Edge Devices*

Xiaolin Li et.al [8], healthcare results were greatly enhanced by using smart wearable bias to cover cases' electrocardiograms(ECG) for real- time discovery of arrhythmias. Deep literacy ways grounded on convolutional neural networks(CNN) have been effectively applied to the discovery of abnormal ECG measures. The computational complexity of current CNN models, still, prevents their use in low- power edge bias. These models generally include a huge number of model parameters, which causes a lot of calculations, memory use, and power consumption in edge bias. In CNN models, network pruning approaches can minimize model complexity at the cost of performance.

*I. Image Classification*

Meriam Dhouibi et.al [9], Convolutional Neural Networks (CNNs) bear high outturn and low inactivity, leading to a high calculation cost. There were many ways in which the diligence that shown a considerable interest in deep literacy(DL) technology. LeNet- 5 being the CNN that was used in the initial stages that contains several pooling layers in addition to the convolutional layers, completely connected layers convolutional layers that wew two in number. GoogLeNet is a deep network with 22 subcastes, whereas AlexNet has five CONV layers and three FC layers. Yet, implementing equally complicated CNN infrastructures is time-consuming and costly. This study investigates approaches for CNN optimization for image bracket that allow us to achieve the same delicacy with varied parameters in order to compress the model and build an optimum armature.

*J. Identifying Locally uncommon bird species*

Ming Zhong et.al [12] , the neural network model ResNet50 was used to categories the sounds of the two raspberry species. The ResNet50 CNN armature consists of a ResNet model with 48 complication layers, 1 Max Pooling subcaste, and 1 Average Pooling subcaste. The original complication and outside pooling are performed using 7 7 and 3 3 kernel sizes, independently, on the RGB images( size 224 224 3). After that, it heaps the remaining blocks. Because of the skip connection of residual blocks, the model may transfer bigger slants to the original layers. These layers can learn just as snappily as the top layers when training deeper networks. A completely connected subcaste and an average pooling subcaste are accordingly present in the network. The ResNet50 model was trained using the Adam optimizer approach with an original literacy rate of 1e- 4 and a decay factor of 1e- 7.

*K. Accelerator for microcontrollers*

Erez Manor and Shlomo Greenberg [14], running inference on small microcontroller-based devices was difficult due to the high resource and computational power requirements of machine learning algorithms. This study proposes application specific hardware accelerators that aid in using machine learning at the edge of a network. These accelerators will help increase performance of the low powered microcontrollers making them effective at processing data. The use of TesorFlow Lite for Microcontrollers model and Neural Processing Unit custom accelerator will help achieve this goal of processing data at the edge. Efficient hardware-software framework is used to integrate Tensorflow and the accelerator.

*L. Pruning based on automata*

Haonan Guo et.al [15], training of DNN's was difficult due to overfitting and gradient vanishing problems that could occur. The paper proposes a way to gradually prune the weights which would in turn improve the stochastic gradient descent. Learning automata which is a reinforcement learning technique which is used to identify the weakly connected weights. The method proposed effectively learns a sparsely connected architecture during the training process.

*M.* Quantization and Deployment

Pierre-Emmanuel Novac et.al [16], optimizing DNN's in a way that allows for easy deployment on edge devices as well as deals with power consumption, memory and real-time constraints. A new framework known as MicroAI is used to help find a solution to the task at hand. The framework is fairly flexible and a

total of three datasets was used for evaluation. A comparison study of proposed framework and the existing embedded inference engine is also done. ARM Cortex-M4F-based microcontroller is used to evaluate the final results.

*N.* Training DNN model using Tensorflow and PyTorch

Arpan Jain et.al [17], performance characterization of Deep Neural Networks on CPU architectures and compare them to frameworks that are optimized for GPU's. The study also compares the effect of multi-process and single-process training of the DNN's to determine which is more useful and more efficient. In the study, it was noticed that Multi-process was up to 1.47x faster than Single-process.

*O. Pruning the channel*

Yihui He [18] introduces a new channel pruning method to help speed up the process of CNN. an iterative two-step algorithm is proposed to help prune each layer of the neural network. LASSO regression-based channel selection and least square reconstruction are the two steps being used to perform the pruning. With the help of these steps and methods, the author tries to achieve faster inference on the deep neural networks. The results show that a model that was pruned using these steps is 5 times faster than the base model with only a 0.3% error increase in the inference. This effectively shows a way to improve and accelerate the deep neural networks.

## IV. CONCLUSION

The survey conducted focused on how the recent advances techniques in deep learning can be applied to improve edge computing applications. With the aid of edge computing, tasks can now be carried out on hardware with little memory. Understanding deep learning with the process of optimization such as quantization, pruning, and weight clustering, can aid with the deployment of the model on edge devices which will allow for certain computations to run and make decisions automatically on devices that are run by microcontrollers and microprocessors. The need to rely on heavy hardware devices will reduce and the possibility of having solutions to complex problems will be on our fingertips.

## REFERENCE


[1] Zixian Zeng, Qingge Gong, Jun Zhang - "CNN Model Design of Gesture Recognition Based on Tensorflow Framework", Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) 2019.

[2] Kristian Dokic, Marko Martinovic, Dubravka Mandusic - "Inference speed and quantisation of neural networks with TensorFlow Lite and Microcontrollers framework", 2020.

[3] Fangxin Wang, Miao Zhang, Xiangxiang Wang, Xiaqqiang Ma, Jiangchuan Liu - "Deep Learning for Edge Computing Applications: A State-of-the-Art Survey", 2020.

[4] Serosh Karim Noon, Muhammad Amjad, Muhammad Ali Qureshi, Abdul Mannan - "Overfitting Mitigation Analysis in Deep Learning Models for Plant Leaf Disease Recognition", International Multitopic Conference 2020.

[5] Shayan Hassantabar, Mohsen Ahmadi , Abbas Sharific - "Diagnosis and detection of infected tissue of COVID-19 patients based on lung x-ray image using convolutional neural network approaches", 2020

[6] Himadri Mukherjee, Subhankar Ghosh, Ankita Dhar, Sk Md Obaidullah, K. C. Santosh, Kaushik Roy - "Deep neural network to detect COVID-19: one architecture for both CT Scans and Chest X-rays", 2020.

[7] Vemula Omkarini, G Krishna Mohan - "Automated Bird Species Identification Using Neural Networks", 2021.

[8] Xiaolin Li, Rajesh C Panicker, Barry Cardiff, Deepu John - "Multistage Pruning of CNN Based ECG Classifiers for Edge Devices",2021.

[9] Meriam Dhouibi, Ahmed Karim Ben Salem, Slim Ben Saoud - "Optimization of CNN model for image classification", 2021

[10] Ranjith M S, Dr. S Parameshwara, Pavan Yadav A, Shriganesh Hegde - "Optimizing Neural Network for Computer Vision Tasks in Edge Devices", 2021

[11] Preeti Nagrath, Rachna Jain, Agam Madan, Rohan Arora, Piyush Kataria ,Jude Hemanth - "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2", 2021.

[12] Ming Zhong , Ruth Taylor , Naomi Bates, Damian Christey , Hari Basnet , Jennifer Flippin, Shane Palkovitz , Rahul Dodhia, Juan Lavista Ferres- "Acoustic detection of regionally rare bird species through deep convolutional neural networks", 2021.
[13] K.Sireesha, Koora Rushika, Padamati Samtha, Korni Venkata Sriharini - "Bird Species Identification Using Deep Learning", 2022.
[14] Erez Manor, Shlomo Greenberg – "Custom Hardware Inference Accelerator for TensorFlow Lite for Microcontrollers", 2022.
[15] Haonan Guo, Shenghong Li, Bin Li, Yinghua Ma and Xudie Ren – "A New Learning Automata based Pruning Method to Train Deep Neural Networks", 2016.
[16] Pierre-Emmanuel Novac, Ghouthi Boukli Hacene, Alain Pegatoquet, Benoît Miramond and Vincent Gripon – "Quantization and Deployment of Deep Neural Networks on Microcontrollers", 2021.
[17] Arpan Jain, Ammar Ahmad Awan, Quentin Anthony, Hari Subramoni, and Dhableswar K. (DK) Panda – "Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters", 2019.
[18] Yihui He – "Channel Pruning for Accelerating Very Deep Neural Networks", 2017.