Algorithmic techniques for resource-aware deep learning

Md Golam Sarwar Murshed
PhD Candidate
Department of Electrical and Computer Engineering

Clarkson™

Main research topics

- Develop resource-aware algorithms for machine and deep learning
- Apply deep learning techniques to various domains, such as computer vision, robotics, biometrics, healthcare, edge devices and resistive computing
- Edge computing to enhance data security and privacy
- Develop a technique to improve the explainability of deep learning
- Deep learning in biometrics
    - developed age-invariant fingerprint segmentation
    - enhanced the performance of an existing fingerprint authentication algorithm
    - developed APIs to improve the security of Internet of Things (IoT) devices
- Currently working with the Department of Homeland Security to develop a better fingerprint segmentation system

Why deep learning matters

- Achieves recognition, classification, and decision-making accuracy at higher levels than ever before
- Outperforms humans in several tasks, like classifying objects in images, playing games, voice generation & recognition, etc
- Ability to process large numbers of features makes deep learning very powerful when dealing with unstructured data

Resource-aware DL for supermarket hazard detection

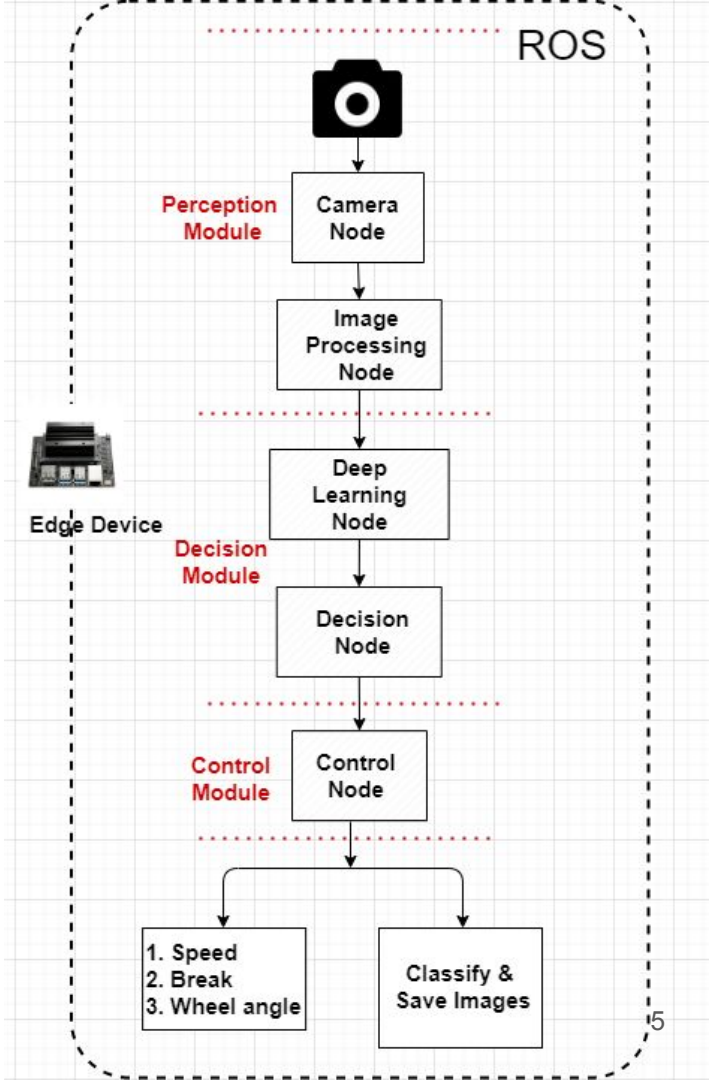
- Supermarkets need to implement safety measures to create a safe environment for shoppers and employees
- Many of these injuries, such as falls, are caused by a lack of safety precautions
- Efforts are increasingly been made to reduce human involvement in hazard detection and inventory maintenance.
- Robots are being used for automated hazard detection and most of this technology involves using cloud servers for data processing
- Resource limitations are a key bottleneck
- Developed a resource-aware DL model, named **Edgelight**, and deployed it on the Marty robot
- This model outperformed the existing models

Fig 1: Autonomous robots patrolling a supermarket (wbur.org)

Fig 2: Images in our supermarket hazards dataset

Application 2: Resource-aware DL on ROS

- EasyDLROS: Developed a novel framework for deploying deep learning on Robot Operating system(ROS)
- A ROS node is a process that performs computations necessary for completing a task
- A ROS topic is a data stream that helps to exchange information between nodes
- Perception module: a camera node capture and process images
- Decision module: Using DL to classify/recognize input images and fuses the results to the other nodes
- Control module: This module contains logics to control the robot



Publications related to resource-aware algorithms

Book Chapters

1. **M. G. Sarwar Murshed**, James J. Carroll, Nazar Khan, Faraz Hussain, **Efficient deployment of deep learning models on autonomous robots in the ROS environment**, Springer, Advances in Intelligent Systems and Computing, 2021

Journals and Conferences

2. **M.G. Sarwar Murshed**, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, Faraz Hussain, **Machine Learning at the Network Edge: A Survey,** ACM Computing Surveys 54, 8, Article 170 (October 2021)
3. **M. G. Sarwar Murshed**, James J. Carroll, Nazar Khan, Faraz Hussain, **Resource-aware On-device Deep Learning for Supermarket Hazard Detection,** 19th IEEE International Conference on Machine Learning and Applications (ICMLA 2020)
4. **M.G. Sarwar Murshed**, Edward Verenich, Conrad Gende, James J. Carroll, Nazar Khan, Faraz Hussain, **Hazard Detection in Supermarkets using Deep Learning on the Edge**, 3rd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 2020)
5. Verenich, Edward, Alvaro Velasquez, **MG Sarwar Murshed**, and Faraz Hussain. "**FlexServe: Deployment of PyTorch Models as Flexible REST Endpoints**." In OpML. 2020.

Mitigating the Class Overlap Problem in Discriminative Localization: COVID-19 and Pneumonia Case Study

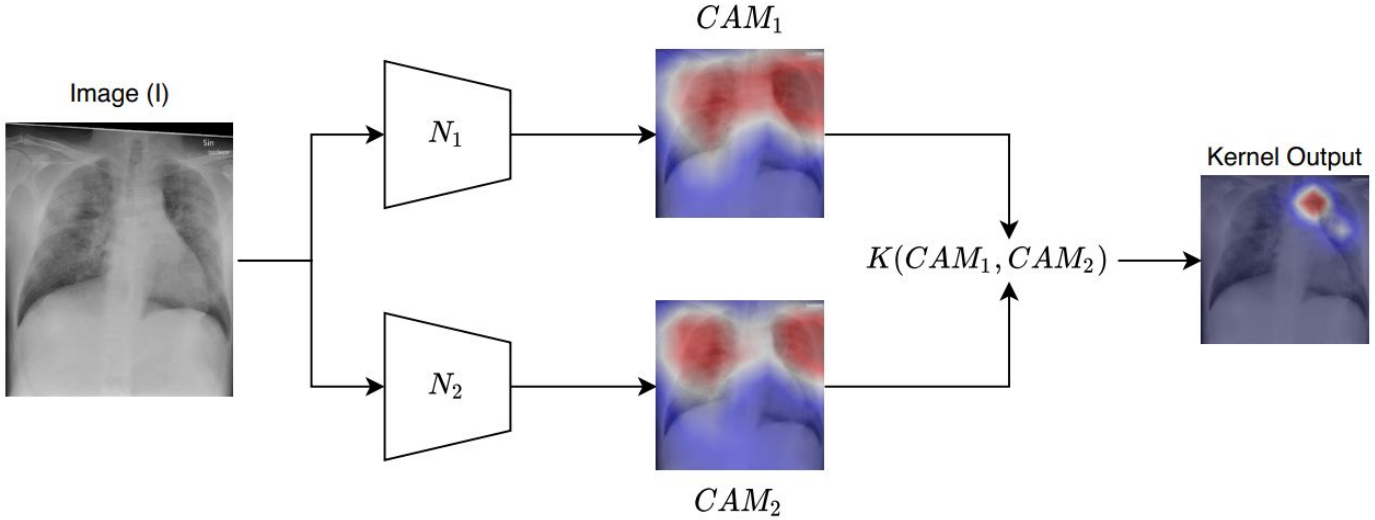


Figure: High-level view of the expert network ensemble architecture. Given a target class c1 and a second possibly overlapping class c2, two binary expert models N1 and N2 are used to calculate class activation maps CAM1 and CAM2 for their respective classes. Class activation maps are then passed to the Amplified Directed Divergence kernel to compute the final class activation map that more narrowly localizes a spatial region associated with CAM1

Book Chapters

Edward Verenich, **M. G. Sarwar Murshed**, Nazar Khan, Alvaro Velasquez, Faraz Hussain, **Mitigating the Class Overlap Problem in Discriminative Localization: COVID-19 and Pneumonia Case Study,** Springer, Explainable AI Within the Digital Transformation and Cyber Physical Systems, 08 May 2021

Fast Resilient-Aware Data Layout Organization for Resistive Computing Systems

- Resistive Computing is a small computing device developing to mimic a synapse in the braina
- It performs energy-efficient multiply and-accumulate (MAC) operations
- It greatly reduces data movement as the computation is performed in-memory
- RCSs are vulnerable to various factors, including non-zero array parasitics
- We propose a framework for fast resilient-aware data layout organization to enable large DNNs to be deployed on RCSs

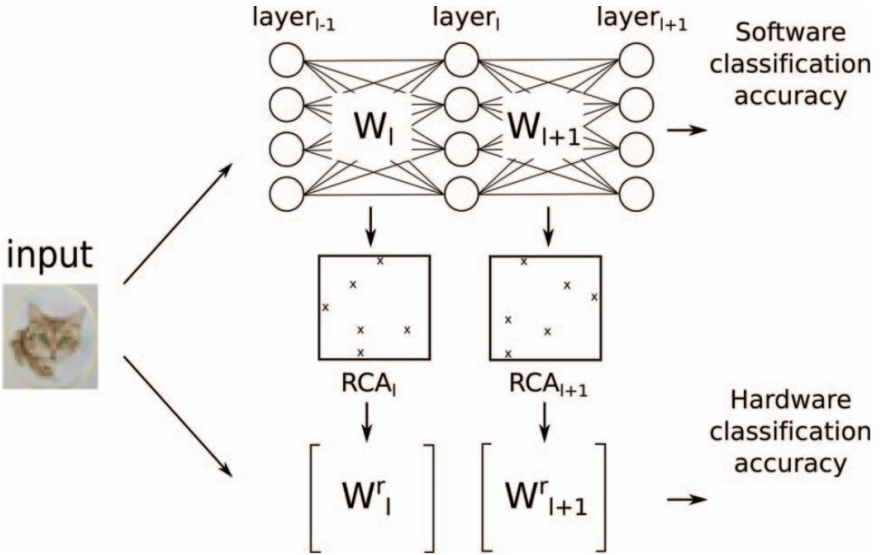


Fig: DNN inference deployment on RCAs of RCSs

Conference paper

Baogang Zhang ; M. G. Sarwar Murshed ; Faraz Hussain ; Rickard Ewetz, **Fast Resilient-Aware Data Layout Organization for Resistive Computing Systems,** 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)

A vision-based system for road crack detection using hybrid deep learning architecture

- Develop deep learning-based road crack detection system
- Used hybrid deep learning based approaches
- One deep learning model is responsible to detect cracks in an input image
- A single-stage object detection YOLO framework is the main deep learning architecture used to detect visual and textual patterns of distress areas
- YOLO is composed of CSPDarknet53 as the backbone, contains 29 convolutional layers 3 × 3, receptive field of 725 × 725
- Another model segment those cracks and calculate the area of cracking regions
- A typical CNN network consists of 9 layers is used in this model
- Random images obtained from the public database were used to test the proposed approach

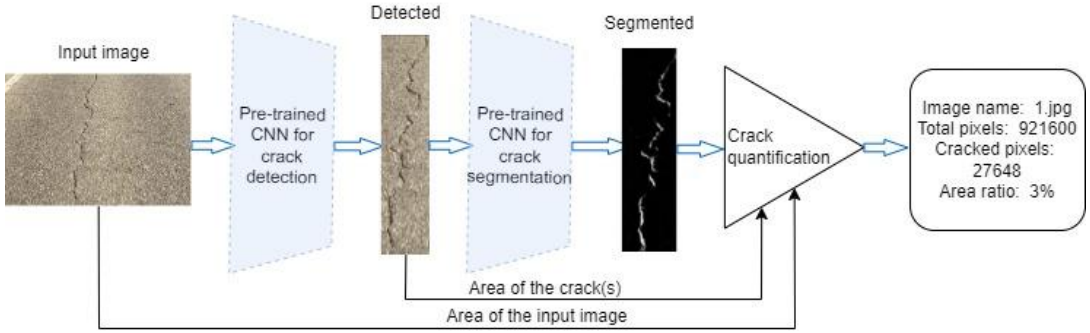


Fig: Overview of Road Crack Detection System

Metric	Formula	Result
Precision	TP/(TP+FP)	0.9444
Recall	TP/(TP+FN)	0.9444
F1 score	2/((1/Recall) + (1/Precision))	0.9433

Deep Age-Invariant Fingerprint Segmentation System

- Develop a new large-scale dataset of contactless finger photos by automatically annotating and labeling the finger images
- Annotation helps to extract the region of interest or finger tips from the finger photos
- Apply different advanced augmentation techniques such as image synthesis [4], varying illumination, flip, zoom, etc to enrich dataset
    - The resulting dataset has 23,650 contactless finger photos with 94,600 fingerprints
- Develop deep learning based finger photo segmentation architecture and train it on our contactless finger photo dataset
- Evaluate and report results using following metrics:
  Matching performance, angle estimation,
  Percentage of segmentations violating the MT

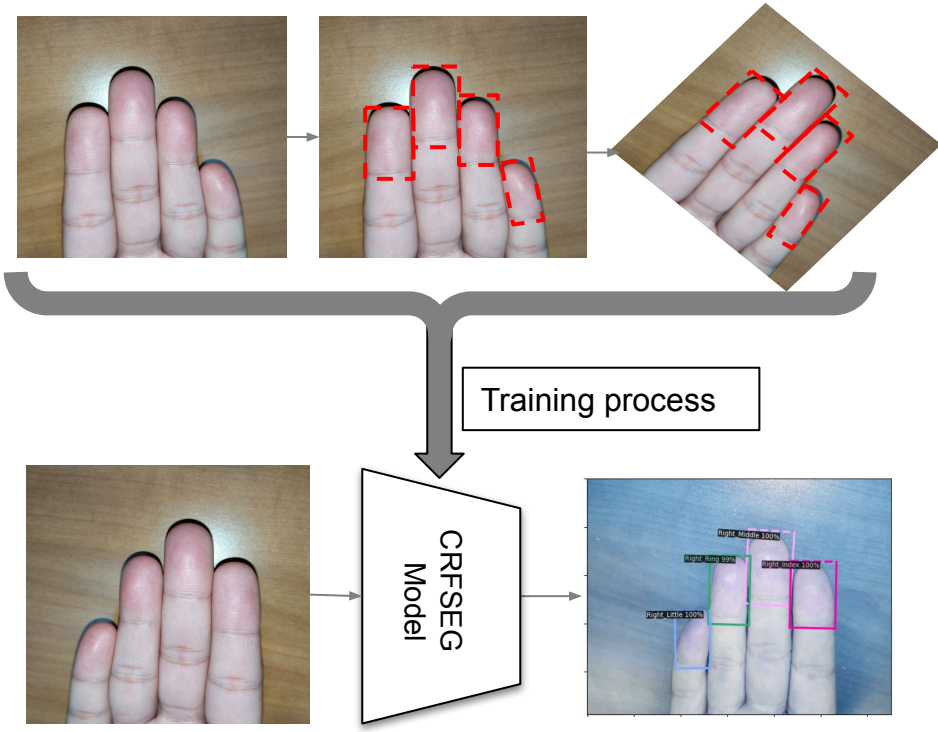


Figure: Develop an automated system that can label and augment finger photos, and train a CRFSEG model using these photos. The trained model can then be applied to other images for segmentation purposes.

[4] L. Dörr, F. Brandt, A. Meyer and M. Pouls, "Lean Training Data Generation for Planar Object Detection Models in Unsteady Logistics Contexts," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019, pp. 329-334, doi: 10.1109/ICMLA.2019.00062.

Deep Age-Invariant Fingerprint Segmentation System

- Develop a new fingerprint detection and recognition algorithm invariant to the over-rotated fingerprints
- Used Faster R-CNN based object detection architecture to detect and recognize slap fingerprints
- Train a deep detection algorithm using our unique dataset containing rotated images
- The resulting algorithm should be invariant to the over-rotated fingerprint and face images. Also performs well on both Juvenile and Adult subjects .
- Evaluate the performance of the new model using following matrices

  Matching performance, Missed Detection Rate (MDR), Fingerprint label accuracy, Percentage of segmentations violating the MT

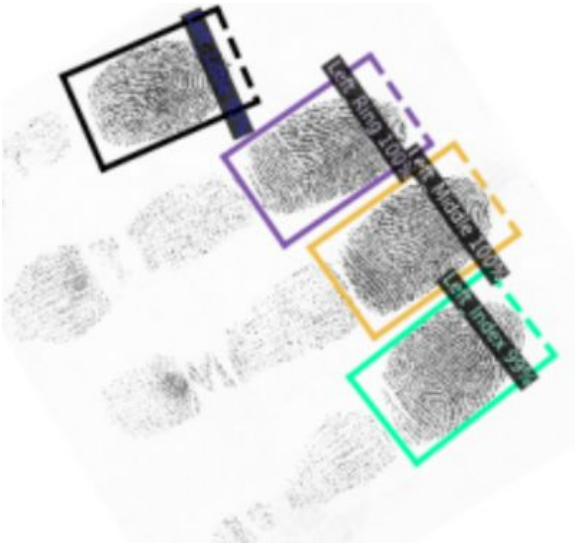


Figure: Output of the DL model which is invariant to the over-rotated fingerprint. It shows the bounding boxes, fingerprint labels and the angle of rotation of the box.

Research related to Cybersecurity

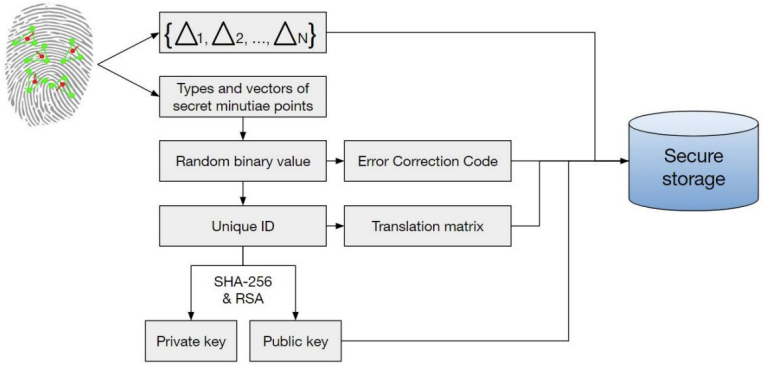
Identity and access management (IAM)

- IAM ensures that only authorized users have access to sensitive data and applications.
- IAM involves managing user **identities, authentication, and authorization**.
- Identity management involves **creating, maintaining, and deleting** user accounts.
- IAM uses technologies such as **multi-factor authentication** and **single sign-on** to enforce security policies.

Authentication using Biometric PKI

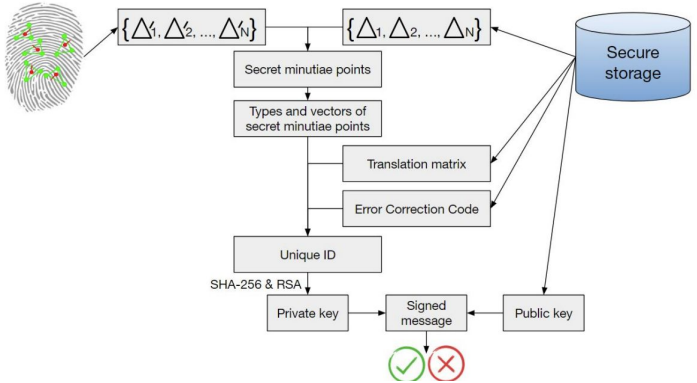
Enrollment

- Scan fingerprint and selects N minutiae points
- Detects 3 closest minutiae points and form triangle.
- Transforms types and vectors of minutiae points and triangle into binary values
- Generates ECC (Error Correction Code) and sends it to Secure storage. can be used further during Matching phase to recover minor data recognition and mapping errors
- Transforms binary values into the Unique ID.
- Unique ID is coded inside Translation matrix.
- Transforms Unique ID into private and public keys using SHA-256 and RSA.
- Everything store in a secure storage



Authentication

- Receives parameters of triangles from Secure storage such as turning angle, length of sides etc and performs searching of triangles
- Having found similar triangles, the authentication process transforms types and vectors of minutiae points and triangle into binary values
- Receives Translation matrix from Secure storage
- Recreates Unique ID using Translation matrix
- generated private key using SHA-256 and RSA derived from the Unique ID
- User identity is verified using the public key stored in Secure storage



Security properties of authentication Systems

Biometric authentication system has three properties which ensure that the system is robust against various types of cyber attacks and threats:

- Non-invertibility: Prevents attackers from using stolen biometric data to impersonate a legitimate user by ensuring biometric data cannot be reconstructed from the extracted template.
- Non-linkability: Prevents attackers from using multiple stolen biometric samples to impersonate a legitimate user by ensuring two biometric samples from the same person cannot be linked.
- Non-repudiation: Provides evidence that a particular user performed a specific action, which helps prevent false accusations against legitimate users.

Detecting malwares from IoT data using deep learning

Objective: To develop an intrusion detection system (IDS) for IoT networks using deep learning techniques

Methodology:

- Clean and preprocess the IoT-23 dataset to remove noise and ensure that the data is in a format suitable for deep learning.
- Develop a deep learning model using convolutional neural network (CNN)
- DL model learns patterns of normal and malicious traffic in the IoT network
- Relevant features used in the project include traffic patterns, packet headers, and payloads.
- Report results using F1 score, Accuracy Rate, Prediction time, Confusion Matrix

Metric	Formula	Result
Precision	TP/(TP+FP)	0.8347
Recall	TP/(TP+FN)	0.8361
F1 score	2/((1/Recall) + (1/Precision))	0.8350

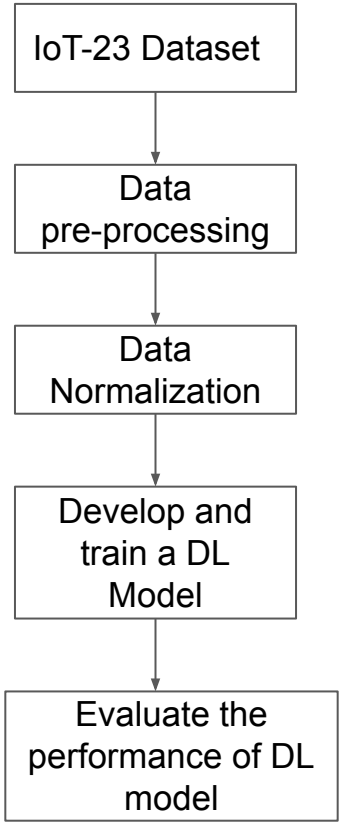


Fig: proposed Malware detection System

Publications of our research

Book Chapters

1. **M. G. Sarwar Murshed**, James J. Carroll, Nazar Khan, Faraz Hussain, **Efficient deployment of deep learning models on autonomous robots in the ROS environment**, Springer, Advances in Intelligent Systems and Computing, 2021
2. Edward Verenich, **M. G. Sarwar Murshed**, Nazar Khan, Alvaro Velasquez, Faraz Hussain, **Mitigating the Class Overlap Problem in Discriminative Localization: COVID-19 and Pneumonia Case Study**, Springer, Explainable AI Within the Digital Transformation and Cyber Physical Systems, 08 May 2021

Journals and Conferences

3. **M. G. Sarwar Murshed**, K. Bahmani, F. Hussain and S. Schuckers**. "Deep Age-Invariant Fingerprint Segmentation System."** arXiv preprint arXiv:2303.03341 (2023)
4. **M. G. Sarwar Murshed,** S. M. Safayet. Hossain, Aksel Seitllari, Kibria K. Roman, **A vision-based system for road crack detection using hybrid deep learning architecture,** ASCE International Conference on Transportation & Development, 2023
5. **M. G. Sarwar Murshed**, R. Kline, K. Bahmani, F. Hussain and S. Schuckers, **Deep Slap Fingerprint Segmentation for Juveniles and Adults**, 2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2021, pp. 1-4, doi: 10.1109/ICCE-Asia53811.2021.9641980
6. **M.G. Sarwar Murshed**, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, Faraz Hussain, **Machine Learning at the Network Edge: A Survey,** ACM Computing Surveys 54, 8, Article 170 (October 2021)
7. M. G. Sarwar Murshed, James J. Carroll, Nazar Khan, Faraz Hussain, **Resource-aware On-device Deep Learning for Supermarket Hazard Detection,** 19th IEEE International Conference on Machine Learning and Applications (ICMLA 2020)
8. Baogang Zhang ; M. G. Sarwar Murshed ; Faraz Hussain ; Rickard Ewetz, **Fast Resilient-Aware Data Layout Organization for Resistive Computing Systems,** 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)
9. Edward Verenich, Alvaro Velasquez, **M.G. Sarwar Murshed**, Faraz Hussain, **FlexServe: Deployment of PyTorch Models as Flexible REST Endpoints**, 2020 USENIX Conference on Operational Machine Learning (OpML 2020)
10. M.G. Sarwar Murshed, Edward Verenich, Conrad Gende, James J. Carroll, Nazar Khan, Faraz Hussain, **Hazard Detection in Supermarkets using Deep Learning on the Edge**, 3rd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 2020)

Thank you!
Q&A

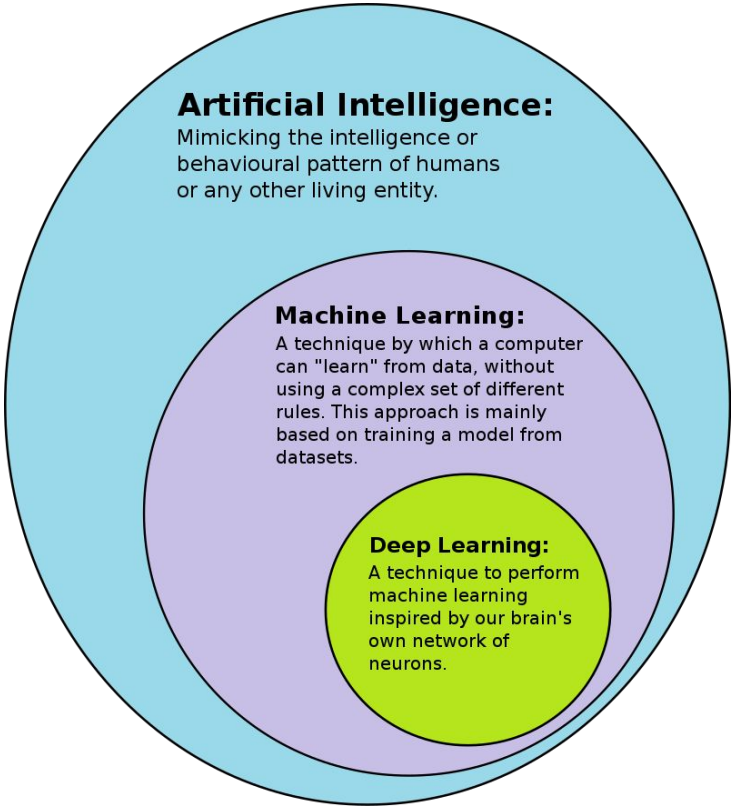
Backup Slides

Outline

- Definition and background of resource-aware deep learning
- Current status of resource-aware deep learning research
- Overview of efficient deep learning and model compression methods
- Our previous resource-aware deep learning works and other case studies
- Challenges in resource-aware deep learning and proposed solutions
- Ongoing and proposed work

What is machine learning? Machine Learning VS Deep Learning

- Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.
- Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modeling.
- Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher-level features from the raw input.
- Machine learning requires a domain expert to identify most applied features. On the other hand, deep learning understands features incrementally, thus eliminating the need for domain expertise.



Background: DL accuracy comes at a huge computational cost

- Deep learning (DL) algorithms have become the most used methods for computer vision, NLP, voice recognition, big data analytics, automated predictions and many other fields
- DL have outperformed state-of-the-art accuracy in most major use cases such as automotive and self-driving cars, voice assistants, drug discovery and toxicology, bioinformatics, medical image analysis, biometrics, etc
- General trend is to make bigger and more complicated networks in order to achieve higher accuracy
- This increases computational cost, memory and power usage
- Certain computing resources such as GPU, and high-speed memory is required for running the traditional DL model
- Mobile computing devices, embedded computing are incapable of fulfilling such computational demand
- Techniques for reducing computational costs, memory, and power requirements of neural networks are highly demanding research areas

How do deep learning algorithms work?

- Most deep learning methods use neural network architectures
- The term “deep” refers to the number of hidden layers in the neural network.
- Networks can have tens or hundreds of hidden layers
- The more the hidden layers are the more computationally expensive it is to operate

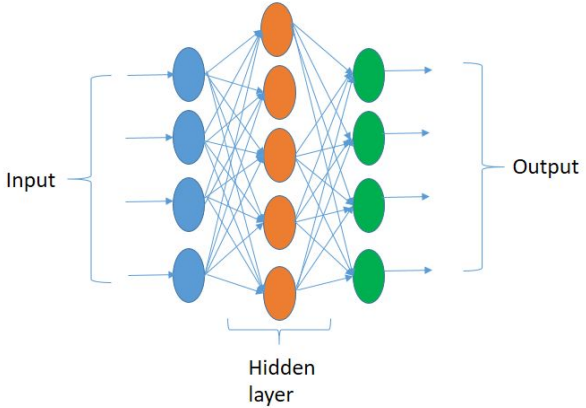


Fig: Artificial neural network

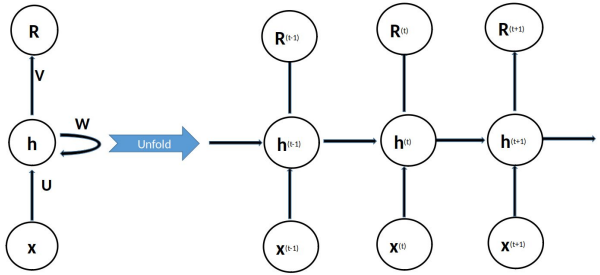


Fig: Recurrent neural network

Convolutional neural networks

- Most popular deep neural networks is known as convolutional neural networks
- CNN automatically extracts features from input data
- First hidden layer could learn how to detect edges
- Last hidden layer learns how to detect more complex shapes specifically catered to the shape of the object we are trying to recognize
- During the DL training process, a CNN algorithm uses the training dataset to learn the features from the data and build a DL model
- The loss function and backpropagation are used to reduce the prediction error of the model.

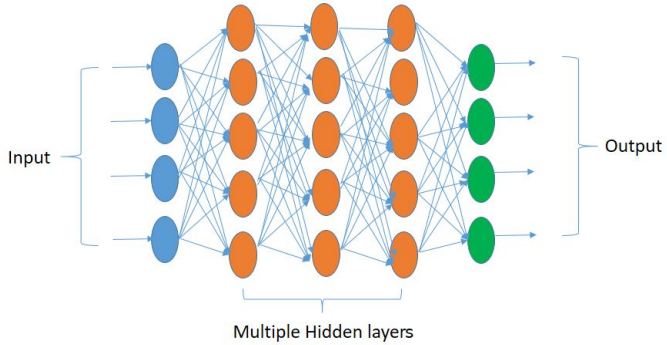


Fig: Convolution neural networks

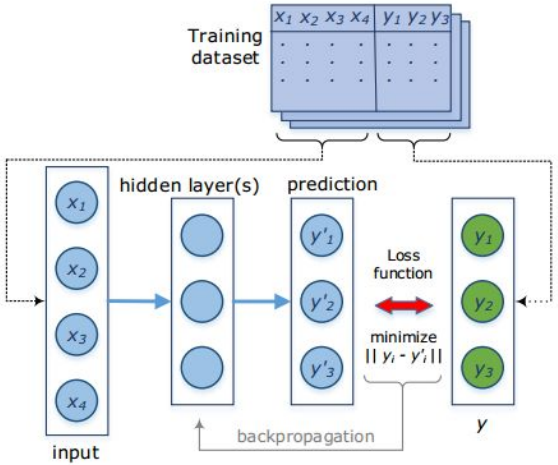


Fig: Deep learning overall training mechanism

img source (http://www.tianyuaninfo.com/?cat=1)

Architecture of Convolutional Neural Networks

1. Convolutional layer
   a. Key component of CNN
   b. Detect the presence of a set of features in the input data
   c. This is done by convolution filtering
   d. The filters correspond exactly to the features we want to find in the images
   e. Filters are automatically learnt by CNN
   f. Several input images need for learning filters
2. Pooling layer
   a. Placed between two layers of convolution
   b. Reducing the size of the features while preserving their important characteristics
3. ReLU correction layer
   a. Activation function
   b. Replaces all negative values received as inputs by zeros
4. Fully-connected layer
   a. Last layer of a neural network and classifies the input data
   b. Applies a linear combination and activation function

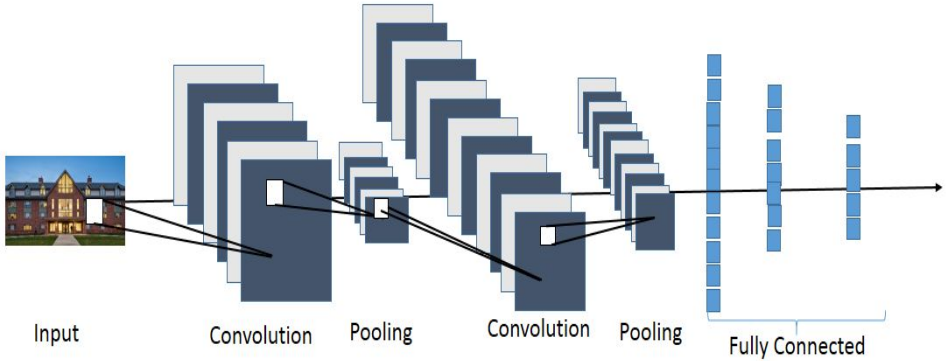


Figure: Basic architecture of a a CNN model

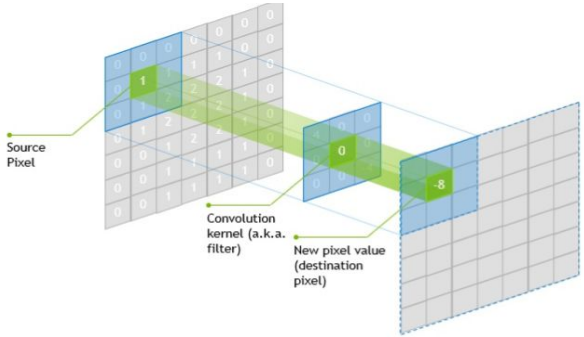


Figure: The convolution operation (source:https://www.embedded-vision.com)

The need for resource-constrained devices

1. Vast majority of modern days applications such as surveillance systems, authentications, patrolling robots, industrial automation and manufacturing, healthcare are used small devices
2. IOT devices, mobile phones, petrol robots, and surveillance camera are continuously producing different types of data
3. These types of devices are very important for real-world applications and industrial use
4. However, these devices are resource-constrained, hance, incapable of processing data and make a decision
5. Potential solutions is transfer data to the cloud for processing which is time consuming, costly, and not secure
6. Deep learning algorithms are state-of-the-art data processing and decision making tools
7. Therefore, the demand for resource-conscious DL algorithms for deploying DL models on resource-constrained devices is increasing day by day.

Current status of Resource-aware DL algorithms

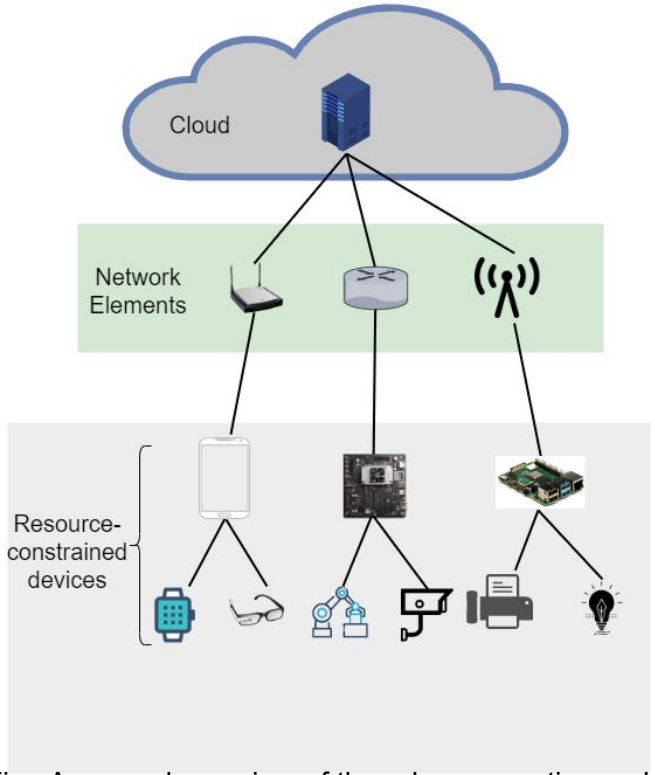


Fig. A general overview of the edge computing architecture: an end-device is one with limited computing capability and some computationally inexpensive work can be performed.Efforts are being made to increase the computing capacity of these devices [9].

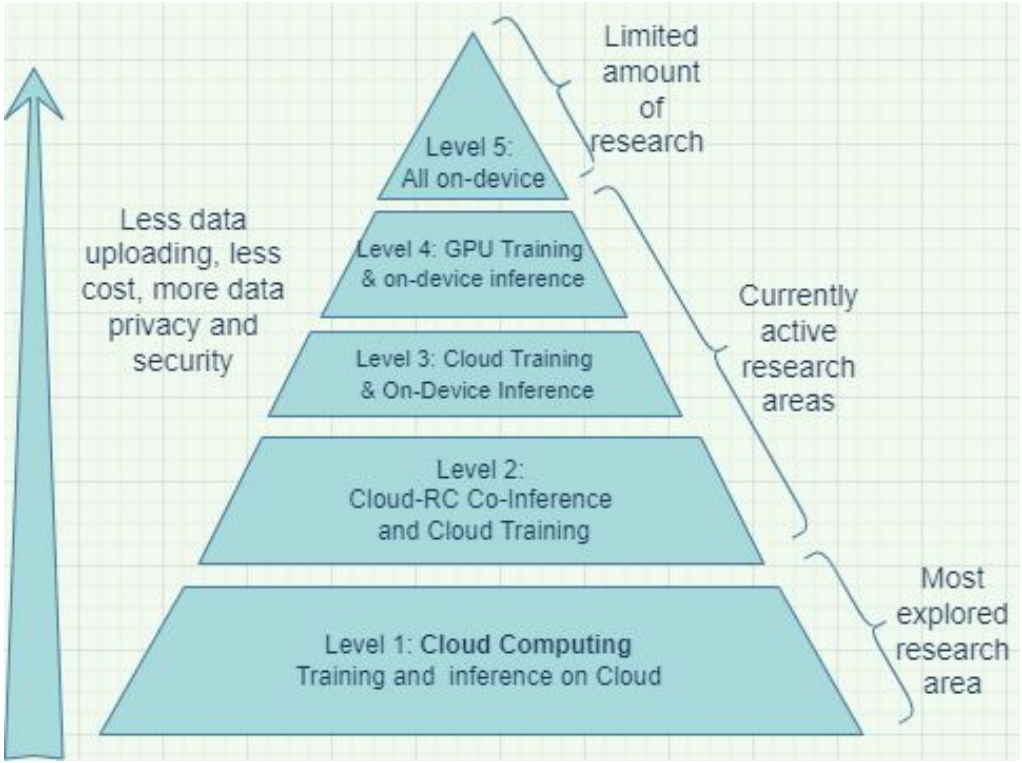


Fig: The current state of research that focuses on deploying deep learning models in resource-constrained devices.

Most popular ways to reduce computation

Model Compression

- Parameter Pruning
  - DL networks are usually over-parameterized
  - Pruning removes the redundant elements in neural networks
  - Reduce the model size and computation cost
  - Common pruning methods: fine-grained pruning, pattern-based pruning and structured pruning
- Quantization
  - Quantize a full-precision weight (32-bit floating-point value) to lower precision (16-bit, 8-bit)
  - Cluster quantization methods apply k-means clustering to find the shared weights of a trained network and weights that fall into the same cluster will share the same weight
  - Linear/uniform quantization directly rounds the floating-point value into the nearest quantized values
  - Different bit-precisions method are greatly reduce the model size but drop accuracy a little bit
  - Quantization-aware training helps to reduce the accuracy loss

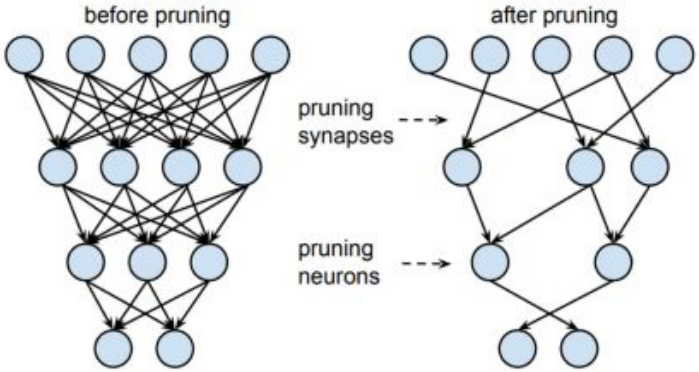


Fig: Parameter pruning [4]

Fig: k-means weight quantization [5]

Most popular ways to reduce computation

Lighter and Faster Architectures

- Depthwise separable convolutions (DSC) provide a lightweight CNN architecture
- A standard CNN model uses each convolutional layer to generate a new set of outputs by filtering and summing the input channels.
- Standard convolutions have the computational cost of:

$$SC = D_k \cdot D_k \cdot M \cdot N \cdot D_p \cdot D_p$$

Kernel size is Dk × Dk,

Output feature map size is Dp × Dp

M input channels,

N numbers of Kernels

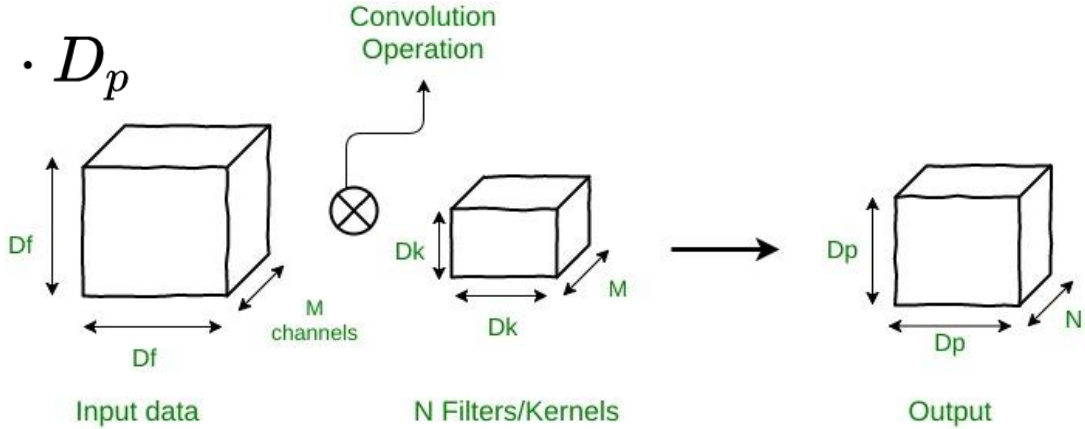


Fig: Computational operation for a standard convolution

Source: https://www.geeksforgeeks.org/depth-wise-separable-convolutional-neural-networks/

Most popular ways to reduce computation

- Depthwise separable convolutions divide each convolutional layer into parts: Depth-wise convolutions, Point-wise convolutions
  - which serve the same purpose as a single convolutional layer
  - greatly reducing the model size and computational cost
- Total number of multiplication for depth-wise convolution

$$DW = D_k \cdot D_k \cdot M \cdot D_p \cdot D_p$$

- Total number of multiplication for point-wise convolution

$$PW = N \cdot M \cdot D_p \cdot D_p$$

- So the total cost DW+PW
- Reduction ratio in computation is = (DW+PW)/SC = $1/N + 1/D_k^2$
- For a K = 5, M = 64, and N = 128 DSC will reduced computational coast by approximately 21 times
- Example of using DSC MobileNets[1], ShuffleNet[2]

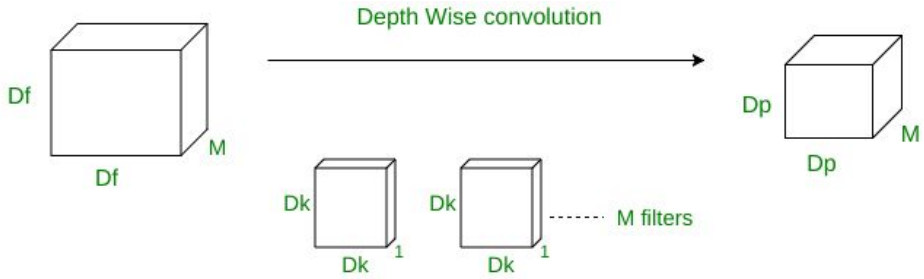


Fig 1. Depth-wise convolution

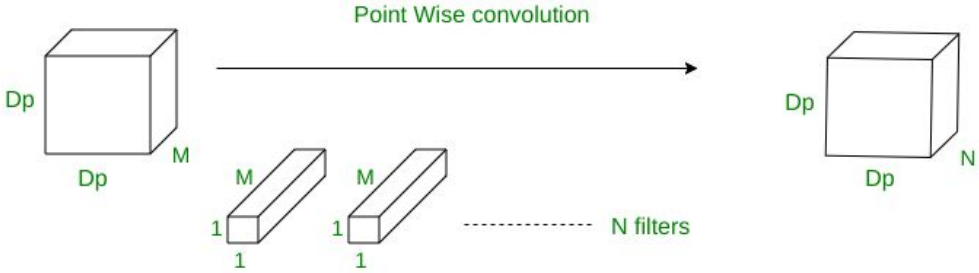


Fig 2: Point-wise convolution

Source: https://www.geeksforgeeks.org/depth-wise-separable-convolutional-neural-networks/

Most popular ways to reduce computation

- Federated Learning
  - Distributed DL methods
  - Involve collaborative training of shared prediction DNN models on devices such as mobile phones
  - Two steps training process, local training and global aggregation
  - Small device download the model from server and computes an updated using local data
  - Server aggregates these updated models
  - Improve data security

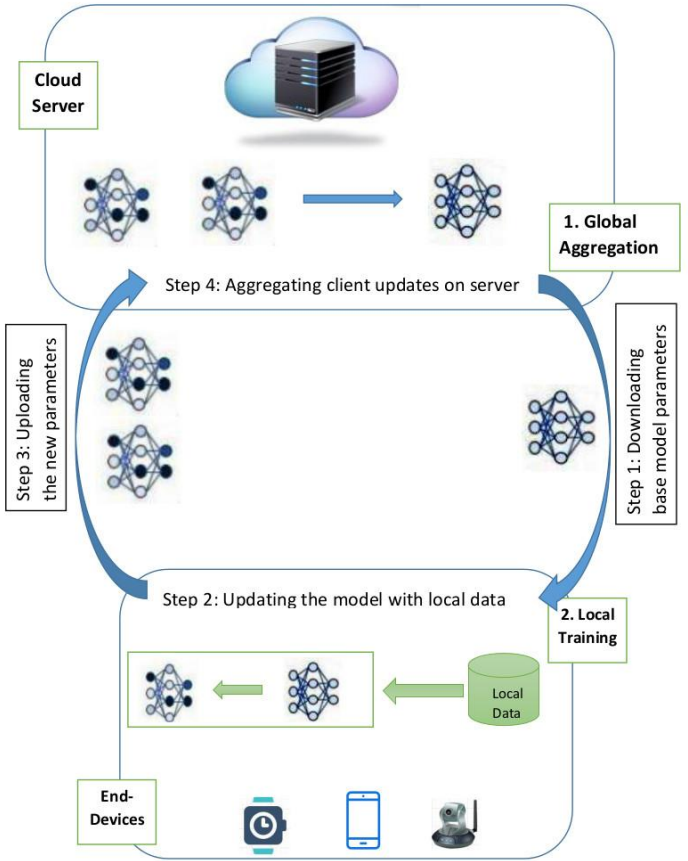


Fig: Federated learning

Resource-aware DL for supermarket hazard detection

- Developed a resource-aware DL model by redesign the googlenet DL architecture
- Used model pruning and quantization to build a deep learning-based model, called EdgeLite, to detect the presence or absence of hazards in supermarket floor
- A new synthesis dataset of supermarket hazards images
- A comparison of EdgeLite with six deep learning(viz. MobileNetV1, MobileNetV2, InceptionNetV1, InceptionNetV2, ResNet and GoogleNet in terms of memory, inference time, and energy
- Open source: https://github.com/sarwarmurshed/supermarket_hazard_detection

Type	Patch size/stride	Output size
Conv	$7 \times 7/2$	$112 \times 112 \times 64$
Max pool	$3 \times 3/2$	$56 \times 56 \times 64$
Conv	$3 \times 3/1$	$56 \times 56 \times 192$
Conv	$3 \times 3/1$	$56 \times 56 \times 256$
Conv	$3 \times 3/1$	$56 \times 56 \times 480$
Pool	$3 \times 3/2$	$14 \times 14 \times 480$
5× EdgeLite_conv		$14 \times 14 \times 832$
Pool	$3 \times 3/2$	$7 \times 7 \times 832$
2× EdgeLite_conv		$7 \times 7 \times 1024$
Pool	$7 \times 7/1$	$1 \times 1 \times 1024$
Dropout		$1 \times 1 \times 1024$
Linear		$1 \times 1 \times 1000$
Softmax	Classifier	$1 \times 1 \times 2$

Table: The outline of the EdgeLite architecture. This architecture is inspired by the googlenet. Parameter pruning is used to reduces the parameters and model quantization is used before deploying on a resource-constrained device.

Performance of EdgeLite Model

Model	GoogleNet	EdgeLite
Size	5.05 (MB)	4.90 (MB)
Parameters	6.02 (millions)	5.26 (millions)

Table: The side and the number of parameters of original model and compressed model

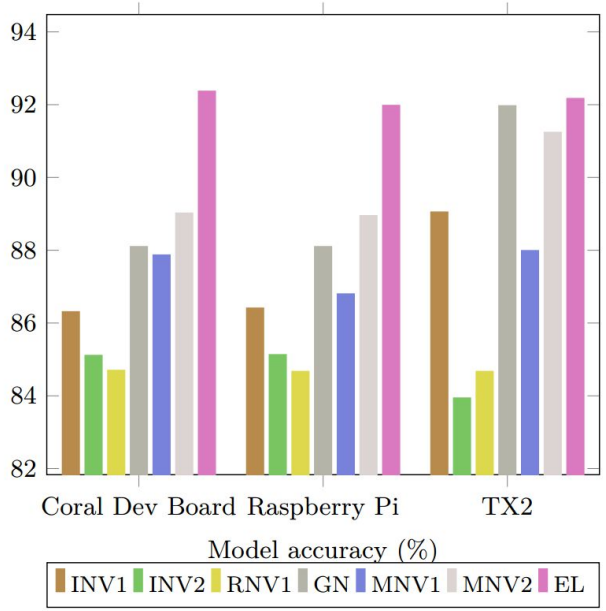


Fig: Model accuracy of different DL models on grocery hazard dataset

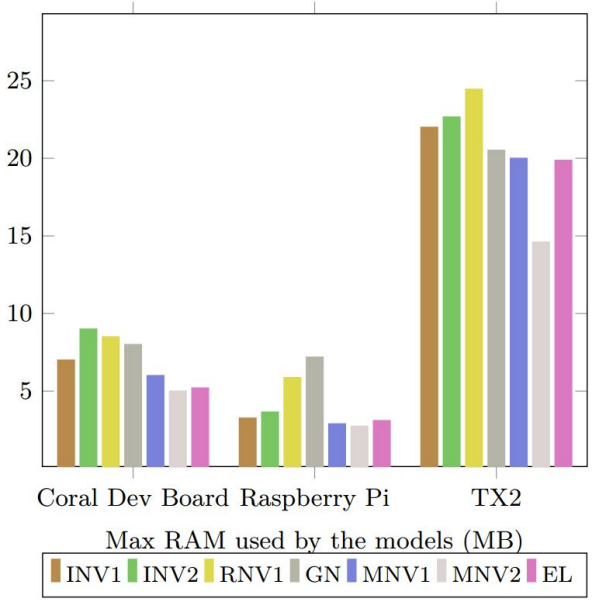


Fig: maximum memory usage at any point during inference

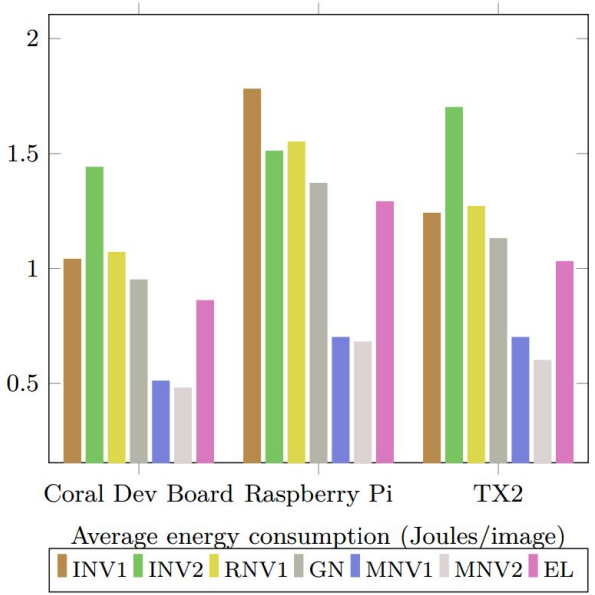


Fig: the average energy consumption (joules/image) during classification

Resource-aware DL on ROS

- The design of EdgeLite, a lightweight image recognition CNN architecture for detecting the presence or absence of supermarket floor hazards
- A novel framework, EasyDLROS, for deploying deep learning models on autonomous robots integrated within the ROS environment
- A comparison of EdgeLite with six state-of-the-art deep learning models (viz. MobileNetV1, MobileNetV2, InceptionNet V1, InceptionNet V2, ResNet V1, and GoogleNet) for supermarket hazard detection when deployed on the NVIDIA Jetson TX2 showing EdgeLite to have the highest F-1 score and comparable resource requirements in terms of memory, inference time, and energy
- A new dataset of images of floor hazards in supermarkets
- A case-study using a real-world example with a robot in a simulated environment.

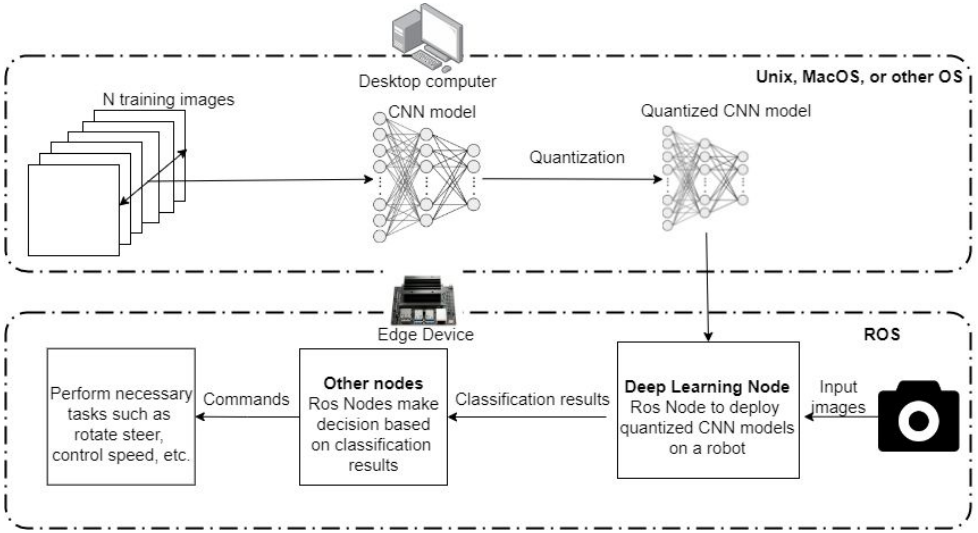


Fig: This figure depicts an overview of our EasyDLROS framework. First, deep learning models are trained on a desktop computer using frameworks such as TensorFlow and MXNet. Then, the models are compressed which helps to reduce compute requirements and facilitate easy integration with ROS. Finally, the models are deployed on the robot and the output results obtained from the DL models can be used to control the robot and perform other tasks.

Results of DL models on ROS environment

Model	F-1 score	Accuracy	Precision	Recall	Avg. inference time(sec)	Max. RAM usage(MB)	Avg. energy used(J/image)
InceptionV1	87.50	88.00	91.30	84.00	0.84	70	**3.18**
InceptionV2	83.35	83.67	82.71	84.00	1.23	120	3.16
ResNet	77.61	79.67	88.14	69.33	1.13	120	3.59
GoogleNet	89.04	89.00	91.54	86.67	0.94	90	3.45
MobileNetV1	87.01	86.68	84.81	89.33	0.34	50	3.41
MobileNetV2	85.53	84.67	80.95	**90.67**	**0.32**	**30**	3.28
EdgeLite	**90.00**	**90.39**	**92.08**	88.02	0.93	70	3.32

Table: The F-1 score, accuracy, precision, recall, average inference time, maximum RAM usage, and average energy consumption of a deep learning model on the Jetson TX2 when operated in a ROS environment using the EasyDLROS framework.

Application: Deep Slap Fingerprint Segmentation for Juveniles and Adults

Problems

- Fingerprints have been used for biometric recognition
- In many applications, 4 fingerprint slaps are used instead of single fingerprint for better identification accuracy
- The fingerprint processing pipeline has to accurately segment the 4 fingerprint slaps into individual fingerprints
- Current fingerprint segmentation algorithms have been trained only on adult datasets
- Compared to adults, juvenile fingerprints have different sizes, spatial characteristics and quality
- This potentially lead to suboptimal fingerprint segmentation which subsequently leads to lower identification accuracy
- Demand to develop new fingerprint segmentation models capable of effectively processing both adults and juvenile fingerprints

Figure: A sample slap fingerprint image

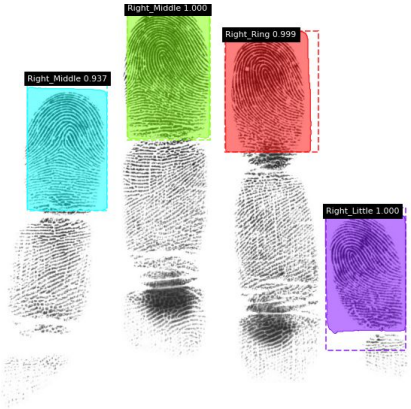


Figure: Fingerprint image segmentation

Application: Deep Slap Fingerprint Segmentation for Juveniles and Adults

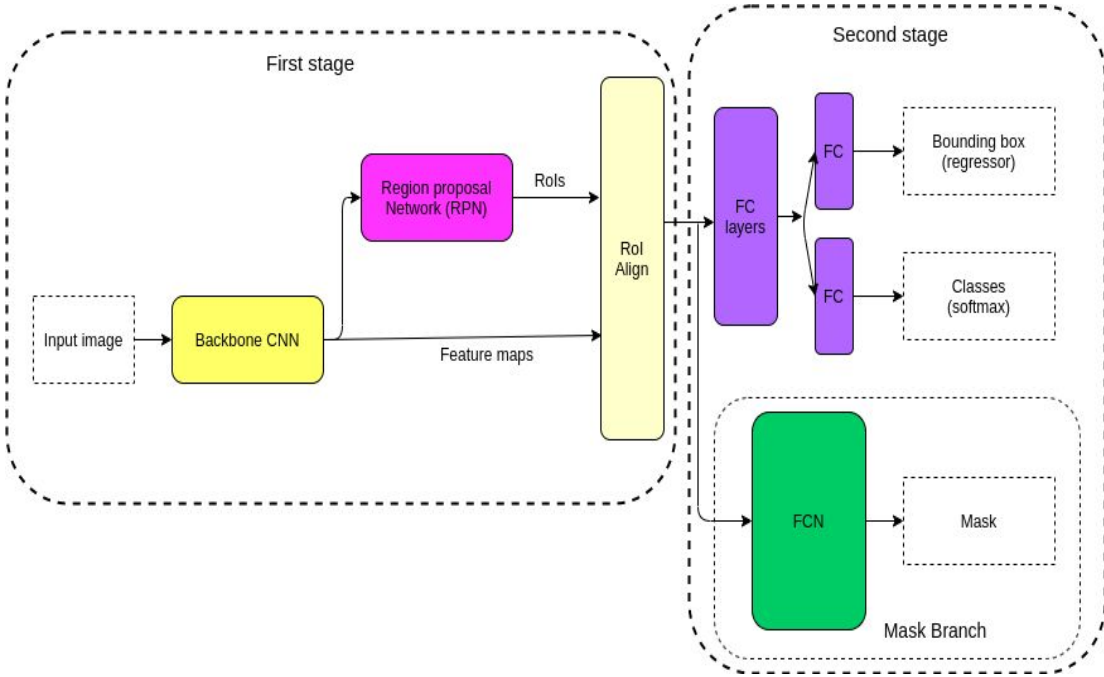
Key contributions

- Developed new tools and used human annotations to establish a ground-truth baseline of this dataset
- Evaluated the combined performance of NFSEG, a well-known benchmark algorithm in our dataset
- Developed a deep learning-based segmentation system, called CFSEG, that is invariant to adult and juvenile subjects
- Compared the performance of CFSEG with the state-of-the-art NFSEG model

Figure: Fingerprints cropped using NIST NFSEG (Subjects are younger than 14 years)

Clarkson Fingerprint Segmentation (CFSEG)

- Based on Mask R-CNN architecture
  - Two-stage deep architecture
  - Developed to perform instance segmentation
- A CNN is the backbone of the Mask R-CNN
  - Responsible for extracting feature maps from the input images
- Region Proposal Network (RPN) proposes candidate object bounding boxes
- Fully Connected layers takes the proposed regions
  - Predicts bounding boxes and object classes
- A fully convolutional network responsible for predicting segmentation masks



Result: Mean Absolute Error

TABLE: Mean Absolute Error (MAE) [Mean (Std.)] for NFSEG and CFSEG

Age Group	Adults		Children	
Model	NFSEG	CFSEG	NFSEG	CFSEG
Left	**07.13(28.59)**	08.21(14.66)	12.09(38.19)	**08.36(16.25)**
Top	13.18(44.70)	**13.05(20.26)**	28.47(92.51)	**13.77(21.13)**
Right	09.46(31.52)	**07.73(13.78)**	13.60(34.55)	**07.21(14.16)**
Bottom	35.23(84.73)	**16.60(24.41)**	102.87(159.00)	**15.00(22.92)**

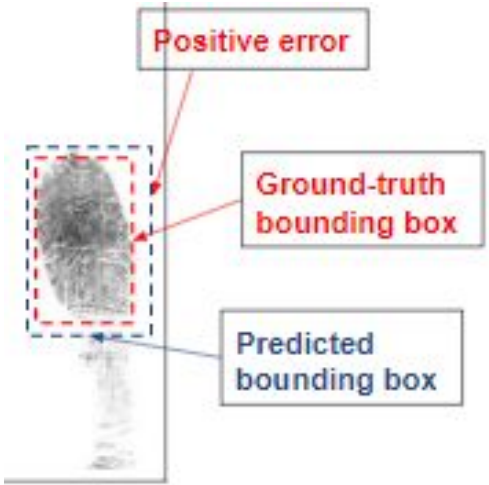


Fig: Mean Abs Error

Histograms of Mean Absolute Error (MEA)

- MAE for top and bottom sides are shown in the histograms
- Blue bars indicates the performance of NFSEG while Orange indicates the performance of CFSEG
- NFSEG struggles to maintains the performance in Juvenile subject
- CFSEG perform significantly better than NFSEG in both adult and Juvenile subjects
- The NFSEG error is high due to over-segmentation in the distal interphalangeal joint

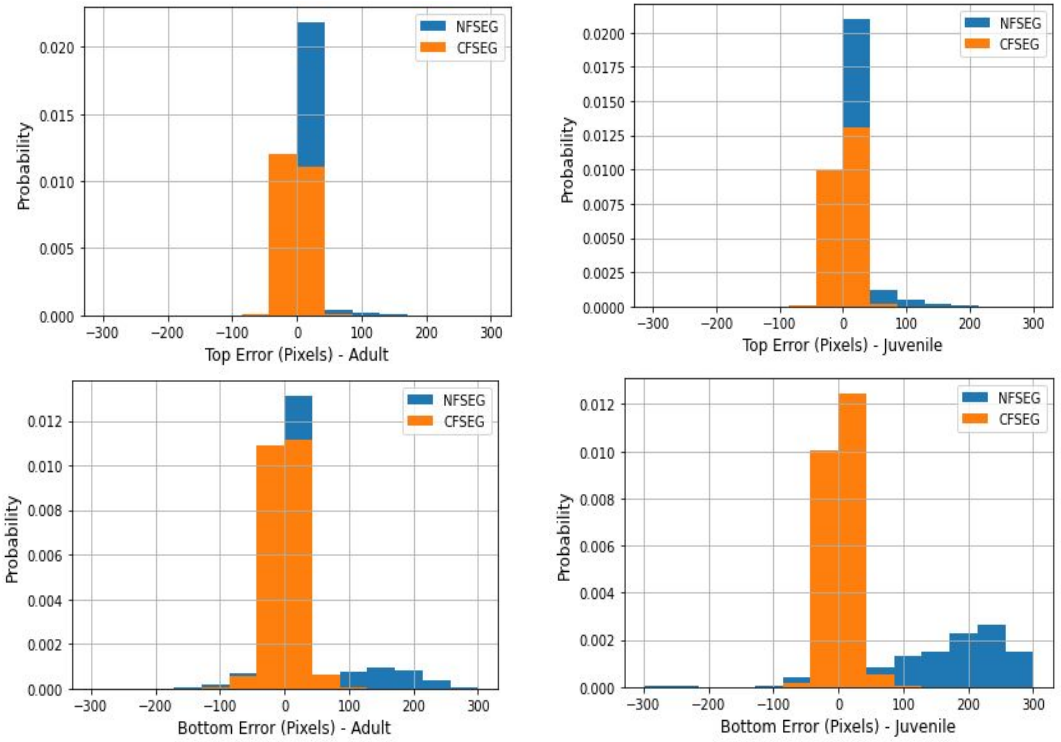


Figure: The histogram of the top and bottom error (pixels) from predicted bounding boxes by the NFSEG and CFSEGmodels for adult (Left) and Juvenile (Right) subjects.

Result: Fingerprint Matching

- The **Verifinger** fingerprint matcher compare the matching accuracy

TABLE 2:TPR [Mean (Std.)] at FPR of 0.001 for NFSEG, CFSEG, and Ground Truth.

Model	Adults	Children
NFSEG	0.9972 (0.0027)	0.9675 (0.0135)
CFSEG	**0.9977 (0.0026)**	**0.9687 (0.0135)**
Ground-Truth	0.9991 (0.0011)	0.9716 (0.0137)

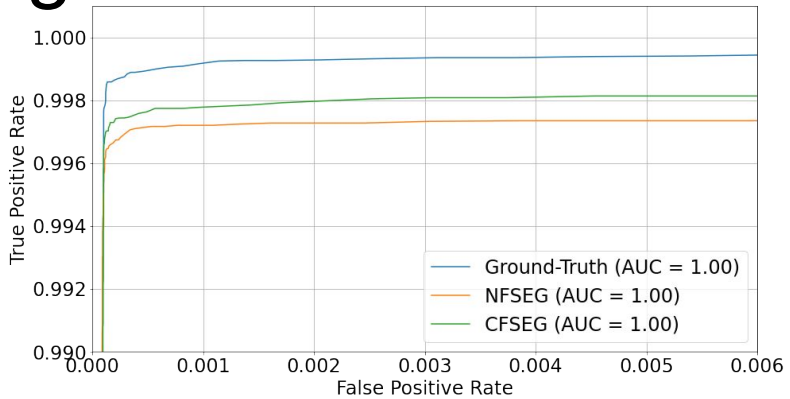


Fig: Receiver Operating Characteristics for for NFSEG,CFSEG, and Ground Truth. (Adults)

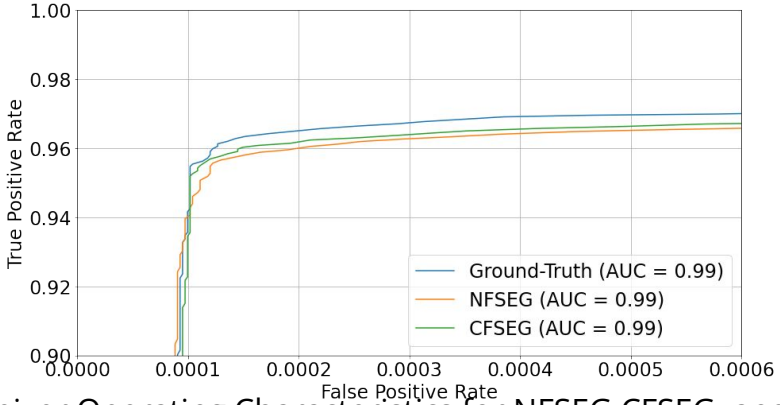


Fig: Receiver Operating Characteristics for NFSEG,CFSEG, and Ground Truth. (Juvenile)

Object detection with arbitrary-Orientation
Using complex deep architectures is challenging

- Existing DL algorithms are heavily agnostic to the object orientation
- Mostly output a horizontal bounding box
- Showed good accuracy when object are straight
- Inappropriate in some real-world settings, where rotation information is critical e.g. aerial objects, biometric trait recognition, etc
- It is crucial to design DL architectures that generate rotated bounding boxes
- Modified R-CNN architecture performs better to detect arbitrary-oriented objects
- However, it consumes huge computational resources, memory and time to train and inference

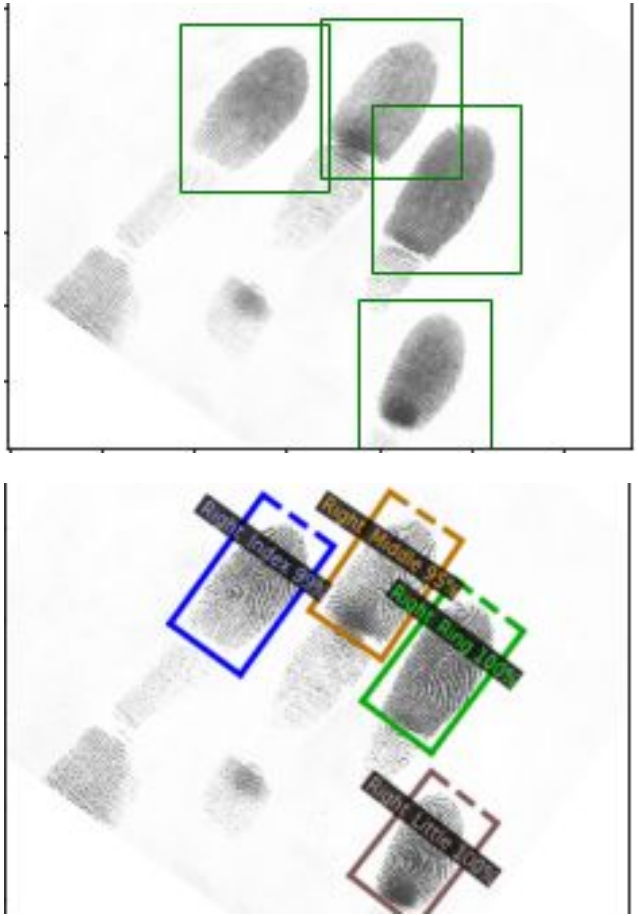


Fig: A straight bounding box do not describe the object outline with high precision but a rotated bounding box

Our Current Method: Arbitrary-Orientation object detection using complex deep architectures

- Develop a new deep learning-based detection and recognition algorithm invariant to the arbitrary-orientated object
- We used over-rotated, and naturally rotated slap fingerprints as arbitrary-oriented objects in this experiment
- Used Faster R-CNN based object detection architecture to detect and recognize slap fingerprints
- Train a deep detection algorithm using our unique dataset containing rotated images
- The resulting algorithm should be invariant to the over-rotated fingerprint images. Also performs well on both Juvenile and Adult subjects .
- Evaluate the performance of the new model using following matrices: Matching performance, Missed Detection Rate (MDR), Fingerprint label accuracy, Percentage of segmentations violating the MT

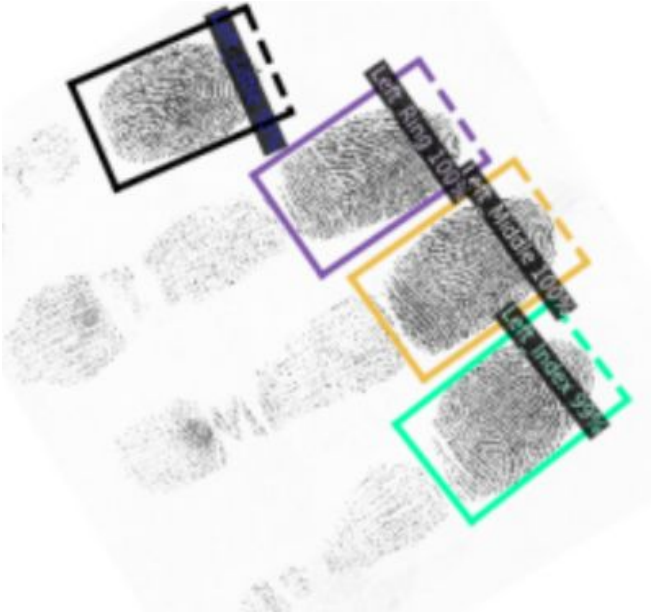


Figure: Output of the DL model which is invariant to the over-rotated fingerprint. It shows the bounding boxes, fingerprint labels and the angle of rotation of the box.

Challenges

- Reduced the computational cost, memory usage, and latency
  - Currently operates on a GPU powered computer with a Core i9 processor
  - Around 30 GB of RAM required for training
  - Inference on GPU: need ~1.5 sec with mask, 0.23 sec without the mask
- Improve the label accuracy
  - Currently we achieved 96.54% on the CU fingerprint dataset
- Improve the matching performance
  - Currently we achieved 98% matching accuracy
  - We forced our model to accurately detected bounding boxes, this has a negative effect on label accuracy
- Reducing computational cost has a negative effect on overall the performance
- Need joint search for best parameters, pruning policy, and quantization policy for better performance

Auto compression and neural architecture search

- Aforementioned model compression strategies rely on hand-crafted heuristics that require domain experts
- This is time-consuming and sub-optimal
- Automated Pruning finds the optimal pruning policy for a given network [6]
  - Different layers in deep networks have different capacities and sensitivities
  - Automated Pruning applies different pruning ratios for different layers of the network to achieve the optimal performance
- Automatic quantization of deep neural networks improves latency, reduces energy usage and maintains good accuracy [7]
  - Automatically determine the quantization policy for a neural network architectures
- Neural Architecture Search (NAS) refers to automatic methods for neural network architecture design [6]
  - Typically requires training and evaluation of thousands of neural networks.
  - Search space and search algorithm are used to address this problem [6]
- It is possible to combine these 3 compression techniques in one which is called “**Joint Compression and Neural Architecture Search”**

Joint search for best parameters, pruning policy, and quantization policy

Problems of using joint architectural and policy search

- Limit opportunities to discover novel primitive operations or building blocks
- The resulting networks sometimes end up being overly complex

Solutions:

- APQ: Joint Search for Network Architecture, Pruning and Quantization Policy [8] proposes to use the quantization-aware accuracy predictor to accelerate this joint optimization
- APQ takes the model architecture and the quantization scheme as input and can quickly predict its accuracy
- Dramatically reduce the search cost
- Active research areas

Planned Contributions

1. Develop a method to reduce the computational cost of deep learning algorithms using advanced automating CNN design and compression techniques such as neural architecture search (NAS), automated pruning and quantization
2. Apply this newly developed compression method to the deep learning-based fingerprint segmentation model and reduce the computation cost of the fingerprint segmentation system
3. Improve state-of-the-art fingerprint segmentation methods to increase the performance of fingerprint matching and labeling accuracy
4. Publish all results and provide all artifacts to enable complete reproducibility and verification.

Milestones

1. Use deep learning (DL) algorithms to solve complex problems such as object detection, recognition, and segmentation **[Done]**
2. Apply manual compression methods, including pruning and quantization on DL models to reduce the computational cost of deep learning models **[Done]**
3. Deploy compressed models on resource-constrained embedded devices **[Done]**
4. Develop a method to reduce the computational cost of deep learning algorithms using advanced automating CNN design and compression techniques such as neural architecture search (NAS), automated pruning and quantization **[Proposed]**
5. Apply this newly developed compression method to the deep learning-based fingerprint segmentation model and reduce the computation cost of the fingerprint segmentation system **[Proposed]**
6. Improve state-of-the-art fingerprint segmentation methods to increase the performance of fingerprint matching and labeling accuracy **[Proposed]**
7. Publish all results and provide all artifacts to enable complete reproducibility and verification **[Proposed]**

Publications of our research

Book Chapters

1. **M. G. Sarwar Murshed**, James J. Carroll, Nazar Khan, Faraz Hussain, **Efficient deployment of deep learning models on autonomous robots in the ROS environment**, Springer, Advances in Intelligent Systems and Computing, 2021
2. Edward Verenich, **M. G. Sarwar Murshed**, Nazar Khan, Alvaro Velasquez, Faraz Hussain, **Mitigating the Class Overlap Problem in Discriminative Localization: COVID-19 and Pneumonia Case Study**, Springer, Explainable AI Within the Digital Transformation and Cyber Physical Systems, 08 May 2021

Journals and Conferences

3. **M. G. Sarwar Murshed**, R. Kline, K. Bahmani, F. Hussain and S. Schuckers, **Deep Slap Fingerprint Segmentation for Juveniles and Adults**, 2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2021, pp. 1-4, doi: 10.1109/ICCE-Asia53811.2021.9641980
4. **M.G. Sarwar Murshed**, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, Faraz Hussain, **Machine Learning at the Network Edge: A Survey,** ACM Computing Surveys 54, 8, Article 170 (October 2021)

Publications of our research

5. M. G. Sarwar Murshed, James J. Carroll, Nazar Khan, Faraz Hussain, **Resource-aware On-device Deep Learning for Supermarket Hazard Detection,** 19th IEEE International Conference on Machine Learning and Applications (ICMLA 2020)
6. Baogang Zhang ; M. G. Sarwar Murshed ; Faraz Hussain ; Rickard Ewetz, **Fast Resilient-Aware Data Layout Organization for Resistive Computing Systems,** 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)
7. Edward Verenich, Alvaro Velasquez, **M.G. Sarwar Murshed**, Faraz Hussain, **FlexServe: Deployment of PyTorch Models as Flexible REST Endpoints**, 2020 USENIX Conference on Operational Machine Learning (OpML 2020)
8. M.G. Sarwar Murshed, Edward Verenich, Conrad Gende, James J. Carroll, Nazar Khan, Faraz Hussain, **Hazard Detection in Supermarkets using Deep Learning on the Edge**, 3rd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 2020)

Publicly available source code

Fingerprint segmenter

- Open source: https://github.com/keivanB/Clarkson_Finger_Segment
- Google Collab: https://colab.research.google.com/github/keivanB/Clarkson_Finger_Segment/blob/main/Clarkson_Fingerprint_Segmentaition.ipynb

Resource-aware DL on ROS

- Open source: https://github.com/sarwarmurshed/supermarket_hazard_detection/tree/master/EasyDLROS

Resource-aware DL for supermarket hazard detection

- Open source: https://github.com/sarwarmurshed/supermarket_hazard_detection

References

1. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. arXiv:1704.04861
2. X. Zhang, X. Zhou, M. Lin, and J. Sun. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6848{6856, June 2018
3. Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015
4. Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both weights and connections for efficient neural networks. In Conference on Neural Information Processing Systems.
5. Dongyoon Han, Jiwhan Kim, and Junmo Kim. 2017. Deep pyramidal residual networks. In IEEE Conference on Computer Vision and Pattern Recognition.
6. Han Cai, Ji Lin, Yujun Lin, Zhijian Liu, Haotian Tang, Hanrui Wang, Ligeng Zhu, and Song Han. 2022. Enable Deep Learning on Mobile Devices: Methods, Systems, and Applications. ACM Trans. Des. Autom. Electron. Syst. 27, 3, Article 20 (May 2022), 50 pages. https://doi.org/10.1145/3486618
7. Coelho, C.N., Kuusela, A., Li, S. et al. Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors. Nat Mach Intell 3, 675–686 (2021). https://doi.org/10.1038/s42256-021-00356-5
8. T. Wang et al., "APQ: Joint Search for Network Architecture, Pruning and Quantization Policy," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 2075-2084, doi: 10.1109/CVPR42600.2020.00215.
9. M. G. Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. 2021. Machine Learning at the Network Edge: A Survey. ACM Comput. Surv. 54, 8, Article 170 (November 2022), 37 pages. https://doi.org/10.1145/3469029

Planned Contributions

1. Developed a method to reduce the computational cost of deep learning algorithms using advanced automating CNN design and compression techniques such as neural architecture search (NAS), automated pruning and quantization on complex DL architectures such as Faster-RCNN
2. Provide resource-aware feature extractor, object detector and segmenter to the research and practitioner community.
3. Improve state-of-the-art fingerprint segmentation methods to improve the performance of fingerprint matching and labeling accuracy
4. Preserving the high accuracy of compressed model
5. Deploy compressed model on resource-constrained settings
6. Publish all results and provide all artifacts to enable complete reproducibility and verification.

Most popular ways to reduce computation

- Lighter and Faster Architectures
  - Depthwise separable convolutions (DSC) provide a lightweight CNN architecture
  - A standard CNN model uses each convolutional layer to generate a new set of outputs by filtering and summing the input channels.
  - Depthwise separable convolutions divide each convolutional layer into two separate layers
    - which serve the same purpose as a single convolutional layer
    - greatly reducing the model size and computational cost
  - Filter of size K × K operating on an input of M and output of N channels, DSC reduces cost by a factor of $\frac{1}{N} + \frac{1}{K^2}$ and the number of parameters being reduced by a factor of $\frac{1}{N} + \left(\frac{M}{K2 \cdot M+1}\right)$
  - For a K = 5, M = 64, and N = 128 DSC will reduced computational coast by approximately 21 times
  - Example of using DSC MobileNets[1], ShuffleNet[2]

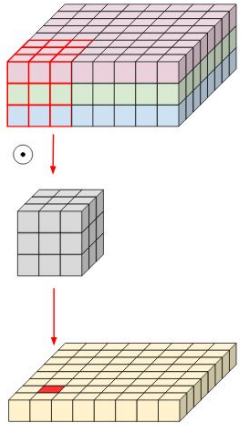


Fig 1. Normal convolution

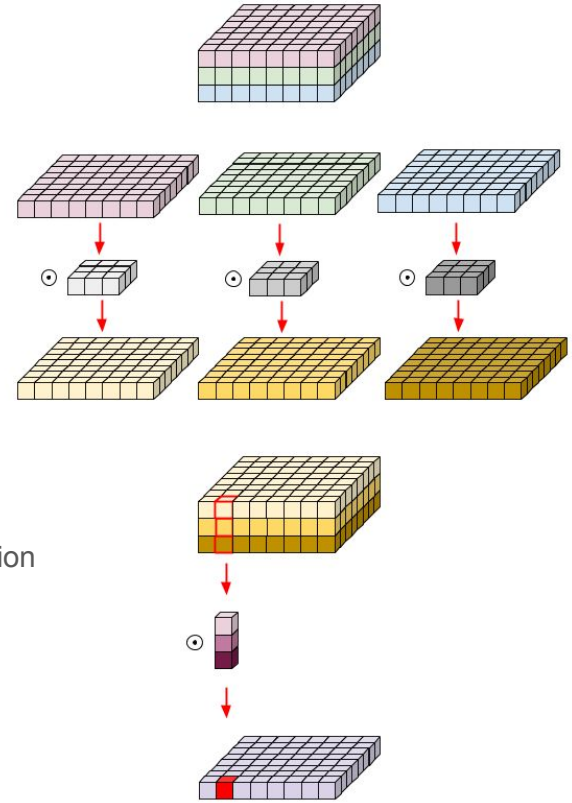


Fig2: Depth-wise separable convolution

Image source: https://medium.com/@zurister/depth-wise-convolution-and-depth-wise-separable-convolution-37346565d4ec

Resource-aware DL on ROS

- 32-bit floating-point format is used in this forward and backward pass.
- Once trained, the model is quantized to 8-bit.
- During inference, the quantized model is used to save computational resources and energy to enable ease deployment on resource-constrained devices.

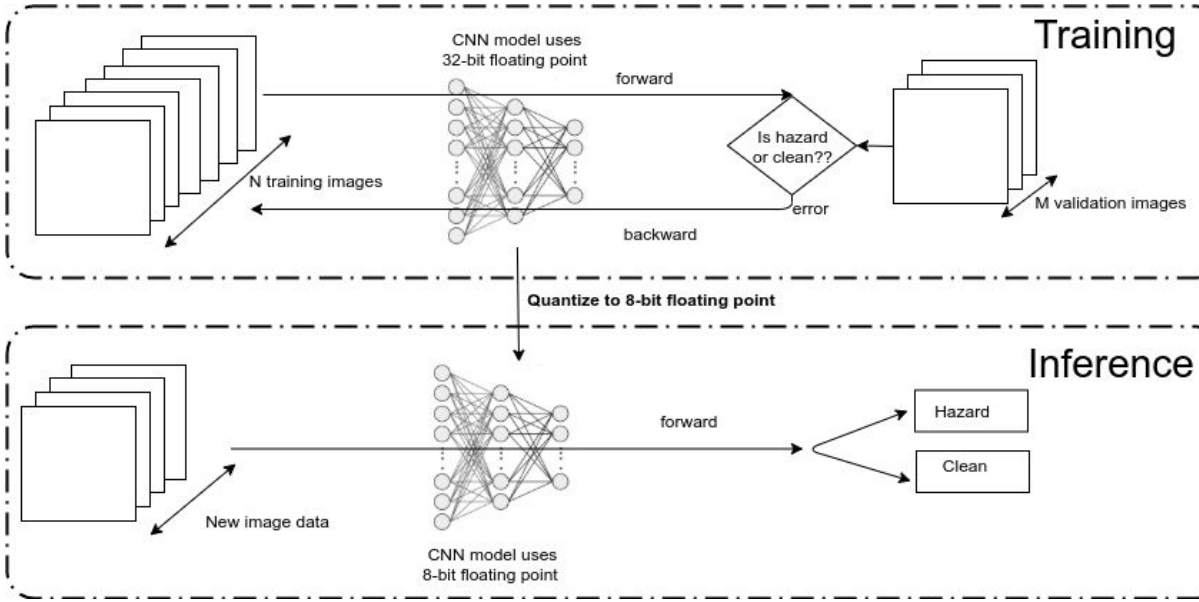


Fig: CNN training and inference for resource-constrained setting. In the forward pass, input data is fed to the network. Each layer of the network uses an activation function to process the input data and calculates an error between the current output vector of the network and the expected output vector. In the backward pass, the network tries to minimize the error by repeatedly adjusting its weights and biases.

More challenging area

- Segment contactless over-rotated fingerprint
- Develop a new large-scale dataset of contactless slap fingerprints by automatically annotating and labeling all slap images
- Introduce rotated fingerprints by utilizing human-annotated bounding boxes
- Developed and Train a deep segmentation algorithm using our unique manually annotated dataset containing both normal and augmented contactless fingerprints

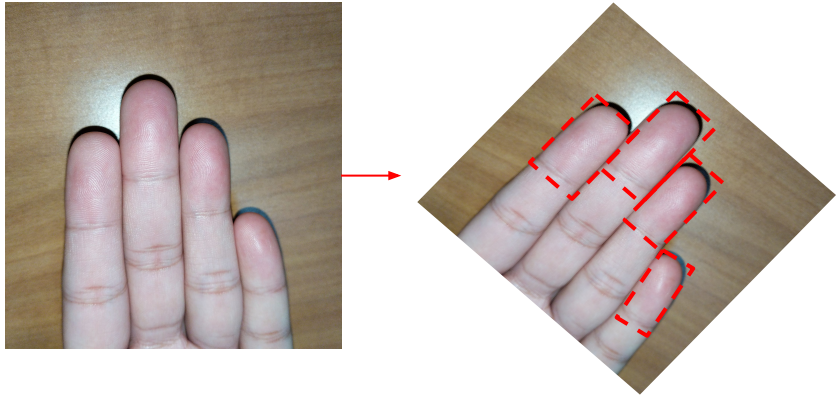


Figure: Draw bounding box around each fingerprint and apply rotation

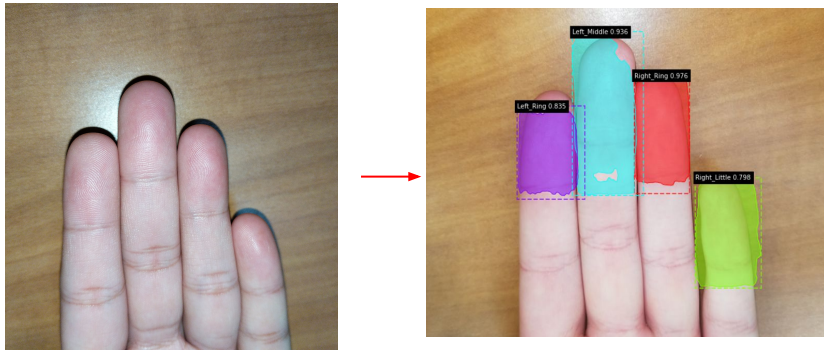


Figure: Rotate fingerprint image with bounding boxes

Experimented with design variations

MobilenetV2 used following variations

1. different feature extractors,
2. simplifying the DeepLabv3 heads for faster computation
3. different inference strategies for boosting the performance

What is new

- MobileNetV3 used image resolutions 96, 128, 160, 192, 224 and 256. The resolutions of our fingerprint images is >1500