FACIAL EMOTION RECOGNITION SYSTEM

Maanasi Shastri (01FE21BCS210), Swarna Patil (01FE21BCS145), Ananya Kulkarni (01FE21BCS144), Amrutha Beedikar (01FE21BCS198)

Under the guidance of
Dr. Sujatha C

KLE Technological University, Vidyanagar, Hubballi-580031, Karnataka, India

June 27, 2024

Overview

- Problem Statement and Objectives
- Dataset Description
- Literature Survey
- Approach
- Experimental Results and Observation
- References

Problem Statement and Objectives

Problem Statement

Develop a deployable Facial Emotion Recognition model on an ultra-low power embedded system.

Objectives

- Design a Deep Neural network-based energy-efficient Facial Emotion Recognition Model.
- Deployment of the Facial Emotion Recognition Model on an ultra-low power embedded device.

Dataset Description[3]

- Benchmark dataset used: "Facial emotion recognition" from Kaggle.
- Dataset size: 56.51MB.
- Grayscale face images.
- The total images in the dataset are 35,914.
- The dataset consists of 7 classes. They are- happy, sad, disgust, angry, fear, neutral, and surprise.
- The size of each image is approximately 2KB.
- Testing images – 20.06% [7,205 images]
- Training images – 79.93% [28,709 images]

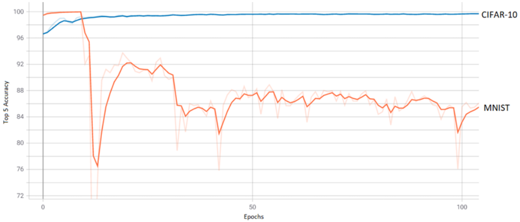
Figure: Sample images from the dataset

- **Data Preprocessing:** Balancing of the training dataset using the "Random Oversampling" method.
  The total number of train images after pre-processing is 50,505.

Literature Survey

1. Benchmarking the MAX78000 artificial intelligence microcontroller for deep learning applications (SPIE, 2022)[2]

- In this study, MAX78000 AI microcontroller's edge-computing deep learning performance and accuracy were assessed. It includes an Arm Cortex-M4F CPU, RISC-V RV32 core, and a 64-channel CNN accelerator with 442KB weight storage and 512KB data memory.
- MNIST utilized ai85net5 (post-QAT Top 1: 67.75%, Top 5: 94.86%), CIFAR-10 employed NAS network (post-QAT Top 1: 89.34%, Top 5: 99.72%). MNIST achieved 99.44% Top 1 and 99.77% Top 5 accuracy, with 1.413 ms inference and 18.3 ms total processing time. CIFAR-10 had 1.715 ms inference time, but accuracy comparison was unavailable.



Literature Survey (Contd.)

2. TinyissimoYOLO: A Quantized, Low-Memory Footprint, TinyML Object Detection Network for Low Power Microcontrollers [4]

- The paper "TinyissimoYOLO" presents an ultra-lightweight, quantized object detection network designed for microcontrollers with limited memory and computational power.
- TinyissimoYOLO supports real-time detection and can handle up to three object classes, demonstrated using the WiderFace and PascalVOC datasets.
- Evaluations show that TinyissimoYOLO achieves up to 73.5% mAP on restricted datasets and outperforms other microcontrollers in energy efficiency and inference speed.

Dataset	Training Restriction	Evaluation Restriction	mAP [%]	Classes
3*WiderFace	None	None	45.3	Faces
	Max 10 obj/image	Max 10 obj/image	46.2	Faces
	Max 5 obj/image	Max 5 obj/image	43.5	Faces
2*PascalVOC	2*Max 3 obj/image	Person	57.4	Person
		Chair	30.2	Chair
		Car	65.1	Car

Table: Performance of TinyissimoYOLO network on WiderFace and PascalVOC datasets.

3. Wildlife Species Classification on the Edge: A Deep Learning Perspective [6]

- The paper proposes an energy-efficient method for real-time classification of wild animal species using a low-power Edge-AI device.
- Experimentation with different deep neural network (DNN) models and image resolutions reveals a remarkable accuracy of 84.30% and an energy efficiency of 0.885 mJ on the MAX78000FTHR board.
- The study explores the impact of image resolution on various performance metrics such as classification accuracy, inference time, memory usage, and energy consumption.

Model	Image Size	Accuracy [%]	F1 Score [%]
VGG-6	64x64	82.88	80.05
	96x96	83.12	82.40
	180x180	84.45	82.67
VGG-8	64x64	81.55	79.67
	96x96	86.67	85.93
	180x180	88.12	86.53

Table: Performance of VGG-6 and VGG-8 models on different image sizes.

Literature Survey (Contd.)

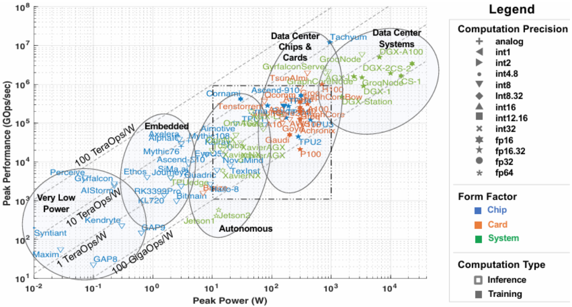
4. Ultra-low Power DNN Accelerators for IoT: Resource Characterization of the MAX78000 (2022)[1]

- The MAX78000 microcontroller stands out by integrating an Arm Cortex-M4F, a RISC-V core, and a dedicated convolutional neural network (CNN) accelerator.
- This study underscores the criticality of optimizing both models and data to achieve energy efficiency, particularly on ultra-low-power devices like the MAX78000.
- It evaluates the MAX78000 ultra-low power DNN accelerator's performance for IoT and wearable applications, analyzing its operational latency, power consumption, and memory footprint across different DNN models, highlighting its potential for battery-powered machine learning applications.

Model	Average Power (mW)	Energy (μJ)
ConvNet5	26.00	215.80
FaceIDNet	32.07	901.17
SimpleNet	25.34	869.16
ResSimpleNet	25.00	787.50
UNet	56.44	3425.91

5. AI and ML Accelerator Survey and Trends(2022)[5]

- The paper outlines the incremental advancements in AI and ML accelerator releases, reflecting significant research and development efforts.
- An architectural overview demonstrates the amalgamation of components from embedded computing, traditional HPC, and HPDA.
- Overall, it underscores the collaborative effort needed to effectively provide AI capabilities for decision makers, warfighters, and analysts.



6. Ultra-Low Power Keyword Spotting at the Edge (2021)[7]

- The paper presents an optimized convolutional neural network (CNN) model for keyword spotting (KWS) designed for energy-efficient deployment on the MAX78000 accelerator.
- The dataset used is called the Speed Command Dataset.
- By eliminating the need for computationally intensive preprocessing steps like Mel-frequency cepstrum coefficients (MFCC), the proposed approach achieves high accuracy and low energy consumption.

Metric	**Value**
Accuracy	96.3%
Energy Consumption per Inference	251 uJ
Number of Classes	12

This table shows key results of the KWS CNN model deployed on the MAX78000 ultra-low-power accelerator.

Approach

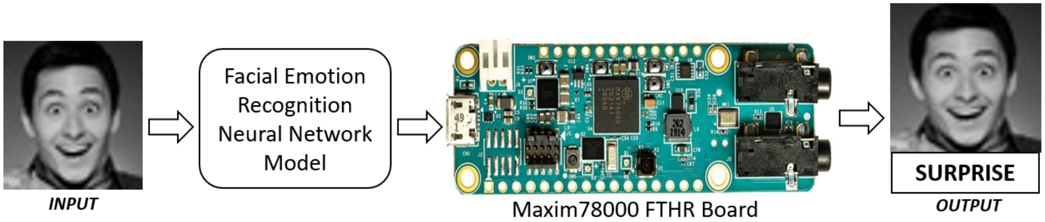


Figure: High level design for FER System

- Model Development: Design the model with PyTorch.
- Training: Train with floating-point weights, then quantize for MAX78000 deployment.
- Model Evaluation: Assess quantized model accuracy using an evaluation dataset.
- Synthesis Process: Use the MAX78000 Synthesizer tool and generate optimized C code from ONNX files, YAML model description, and input data.

Deployment Board: MAX78000FTHR BOARD

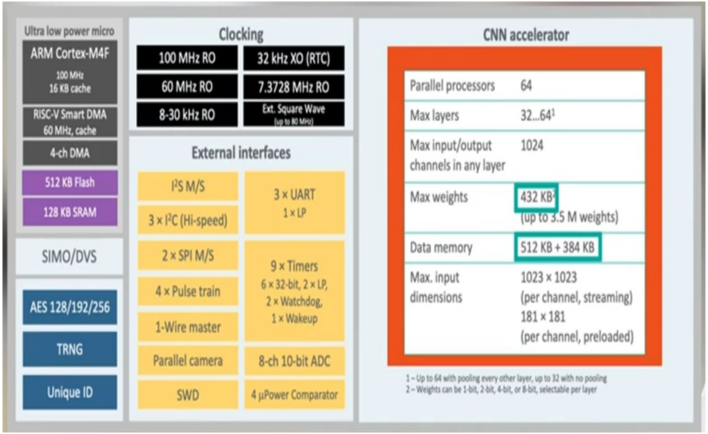


Figure: Deployment Board

Deployment Flow

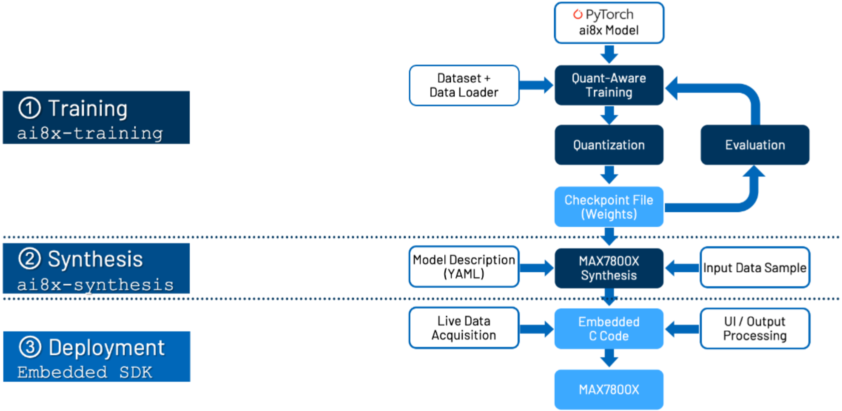


Figure: Deployment Flow

Circuit Diagram to calculate inference energy

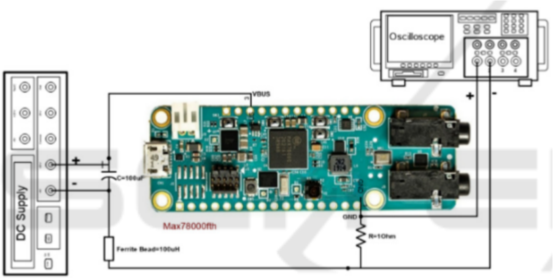


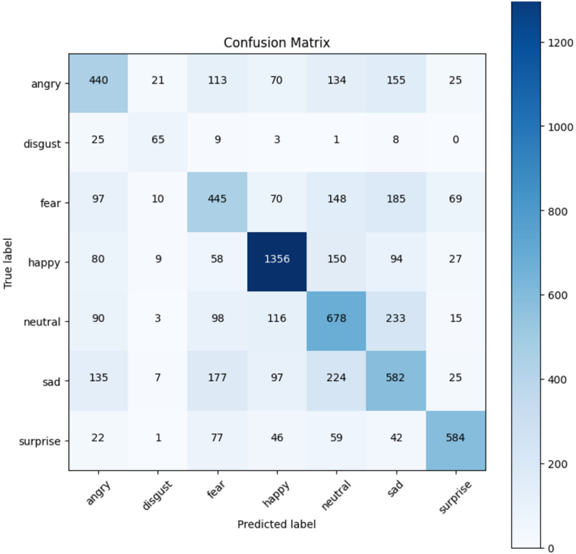
Figure: circuit diagram

Inference Energy = I * V * inference time

I: Current (mA)

V: Voltage (V)

Results and Observations



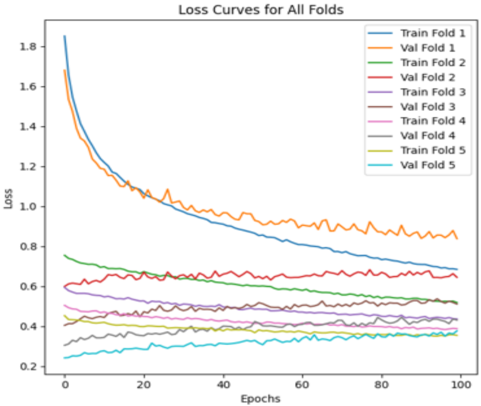
No.of epochs : 100

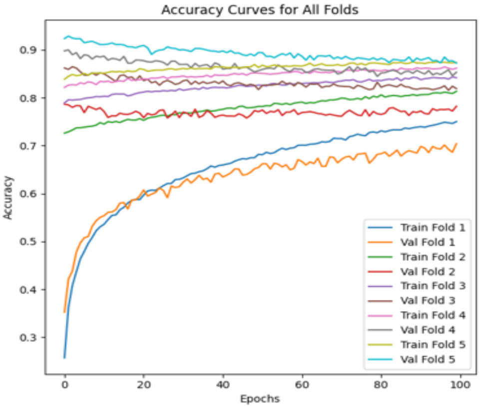
Learning rate : 0.001

Accuracy : 57.82%

Class	Test Accuracy (%)
Happy	77.00%
Surprise	72.44%
Disgust	53.15%
Angry	50.73%
neutral	47.04%
Sad	44.59%
fear	39.84%

Experimental Results and Observations (contd.)





The loss decreases from 2.0365 to 0.4312, indicating improved model convergence. Training accuracy improves from 23.46% to 85.49%, while validation accuracy rises from 37.91% to 46.52%, showing consistent performance.

Experimental Results and Observation (Contd.)

Figure: input image

```
Waiting...

*** CNN Inference Test ai85-fer ***

*** PASS ***

Approximate data loading and inference time: 1521 us

Classification result:
[  14073] -> angry
```

Figure: output

Experimental Results and Observation(Contd.)

Paramenter	**cat-dogs**	**FER**
Dataset size(no.of image in the train set)	25000	50000
Number of classes	2	7
Image size(in pixels)	64x64	28x28
Inference Time (in μs)	12055	1516

Experimental Results and Observation (Contd.)

Resource Usage

Weight memory: 70,572 bytes out of 442,368 bytes total - 16.0%

Bias memory: 7 bytes out of 2,048 bytes total - 0.3%

Inference Energy

Inference Energy = I (mA) * V (V) * inference time (μs)
Inference Energy = 3.16 * 5 * 1.516
Inference Energy = 0.0237 mJ

Reference I

Author PictureLei Xun Author PictureChulhong Min Author PictureFahim Kawsar Author PictureAlessandro Montanari Author PictureArthur Moss, Author PictureHyunjong Lee, 20232, month = 11, pages = 934–940, title = Ultra-Low Power DNN Accelerators for IoT: Resource Characterization of the MAX78000 url =https://doi.org/10.1145/3560905.3568300.

Mitchell Clay, Christos Grecos, Mukul Shirvaikar, and Blake Richey.
Benchmarking the MAX78000 artificial intelligence microcontroller for deep learning applications.
In Nasser Kehtarnavaz and Matthias F. Carlsohn, editors, *Real-Time Image Processing and Deep Learning 2022*, volume 12102, page 1210207. International Society for Optics and Photonics, SPIE, 2022.

FER Dataset from Kaggle.
Facial emotion recognition dataset.
Available at: `https://www.kaggle.com/datasets/chiragsoni/ferdata/data`.

Julian Moosmann, Marco Giordano, Christian Vogt, and Michele Magno.
Tinyissimoyolo: A quantized, low-memory footprint, tinyml object detection network for low power microcontrollers, 05 2023.

Albert Reuther, Peter Michaleas, Michael Jones, Vijay Gadepally, Siddharth Samsi, and Jeremy Kepner.
Ai and ml accelerator survey and trends.
In *2022 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, September 2022.

Mateo Avila Pava Harshit Vaishya Yazan Tabak Juergen Ernst Ruben Portas Wanja Rast Joerg Melzheimer Ortwin Aschenborn Theresa Goetz Subodh Ingaleshwar, Farid Tasharofi and Stephan Goeb.
Wildlife species classification on the edge: A deep learning perspective.
In *16th International Conference on Agents and Artificial Intelligence (ICAART)*, 2024.

Mehmet Gorkem Ulkar and Osman Erman Okman.
Ultra-low power keyword spotting at the edge, 2021.

Thank You