An Efficient Sarcasm Detection in Audio Using Parameter-Reduced Depthwise CNN

Jiby Mariya Jose[1], Jeeva Jose[2]
Independent Researcher

***Abstract*—In this study, we implement a lightweight CNN for sarcasm detection using audio input. To achieve this goal, we propose DepthFire block. We propose a lightweight version of the traditional Depthwise convolution layer that focuses on reduced memory. Unlike the traditional depthwise convolution layer that focuses on reducing the memory requirements of the entire architecture, our solution offers a specific and targeted approach that specifically reduces the memory requirements of the depthwise convolution layer through parameter reduction.We evaluated the impact of its energy consumption and the performance of our pro- posed solution with other existing solutions and on different activations, pooling functions and datasets.We further tested the applicability of the solution on 2D input.And our solution obtained 82.98 percent model size reduction as compared to MobileNetV2 and 58.94 percent as compared to MobileNetV3 small with a energy reduction of 56.48 percent on CIFAR10 dataset.**

***Index Terms*—Energy-Efficient CNN Design, Lightweight CNN Design, Sarcastic Speech Emotion Recognition, Deepfake-Detection, Brain-Tumour Detection, Energy-Efficiency, Reduce Training Time.**

1 INTRODUCTION

Communication stands as a profound bridge that connects living beings with the inanimate, intricately weaving a tapestry of human existence through emotions and understanding [1]. Its significance extends beyond data transmission to encompass deep human interactions and comprehension, grappling with fundamental questions of why, how, and what during data exchanges between entities [2]. In a world where machines are deeply integrated into every facet of human life, the need to interpret human emotions becomes increasingly critical [3]. Consequently, automated emotion and sentiment recognition have garnered significant attention [4]. Sarcasm, a nuanced facet of human emotion, presents a formidable challenge for automated emotion recognition systems [5]. Unlike more straightforward sentiments, sarcasm relies heavily on various factors, such as the speaker's past experiences, current sentiments or emotions, and the contextual environment [6]. Moreover, the listener's ability to comprehend the intended meaning further complicates sarcasm detection [7]. Sarcasm often serves as a means to perform critical communication by making less impact on the listener [8, 9]. However, the success of sarcasm as a communicative approach hinges greatly on the interplay between the speaker, the listener, and the surrounding context [10, 11]. As technology evolves[12], the ability to accurately detect sarcasm through automated means becomes increasingly vital for fostering seamless human-machine interaction and advancing the development of emotionally intelligent systems [13].However, the current state-of-the-art predominantly applies to digital communication and lacks research in direct speech-only-based studies[14].

Deep Neural Networks (DNNs) have emerged as powerful tools for processing various types of data, such as images, videos, language, and speech [15–18]. However, they are known to be computationally demanding during both training and inference [19–22]. A significant portion of these computations involves convolutional operations, which are fundamental in many DNN architectures, especially convolutional neural networks (CNNs) [23, 24]. As CNNs continue to be widely used, finding ways to reduce the computational cost associated with them remains a crucial area of research [20]. The complexity of convolutional operations scales with $O(N^2 \times K^2)$

when applied to an
input of size N with a kernel size K [25–27]. The growing number of convolutions [28]
and multiply-and-accumulate computations (MAC) increases model size, leading to higher demands on CPU and memory resources, thereby extending device runtime [29, 30]. These escalating resource requirements[31] not only slow down computations but also increase energy consumption, presenting significant challenges during train- ing that involve intensive tensor operations in both forward and backward passes [32]. As a result, specialised hardware is increasingly required to support these computations, leading to higher implementation costs [33].
Amidst the growing computational demands of Multiply-and-Accumulate (MAC) operations in conventional convolutional neural networks (CNNs), researchers are exploring more efficient alternatives to reduce the burden on hardware and expedite training and inference times [23, 24, 34, 35]. These strategies involve various optimisations such as reducing the number of network components like neurons, filters, and connections, as well as optimising the size of convolution kernels, down-sampling methods, and precision adjustments for weights and activations [34, 36–38]. Notable examples include EfficientNet, which balances network depth, width, and resolution to reduce complexity; GoogleNet, which focuses on expanding network width; and MobileNet, which implements depthwise separable convolutions to streamline opera- tions. MobileNetV2 enhances efficiency with inverted residuals and linear bottlenecks, while MobileNetV3 further reduces model size using Neural Architecture Search (NAS) combined with depthwise convolution [39, 40]. Despite these advancements, achieving successful and energy-efficient training on CPUs and resource-constrained devices[41] remains challenging due to significant computational demands[42] and power require- ments [43, 44].
To address the challenges outlined above, this paper introduces an energy-efficient and lightweight convolutional neural network (CNN) design that optimises depthwise con- volution operations through multiply-and-accumulate (MAC) reduction for sarcasm detection application.

The following challenges were identified in the application:

- Insufficient Dataset: Existing automatic sarcasm detection models that rely on audio data often utilise datasets such as WITS [45], MUSTARD [14], and MaSaC [46], which are primarily designed for analysing the speech through textual content extraction. However, these models encounter difficulties in accurately capturing the audio patterns associated [47] with sarcasm. This challenge arises due to the lack of a specialized dataset tailored specifically for sarcasm detection from audio inputs [48].
- Limited Edge-Based Solutions: There is a scarcity of edge-based solutions for sarcasm detection based on CNN, which could facilitate real-time processing and inference directly on devices with limited computational resources [49, 50].The absence of such solutions restricts the practical deployment of sarcasm detection systems in resource-constrained environments.
- Neglect of Energy Efficiency: With the increasing concern over energy consumption

in deep learning models, particularly in resource-constrained environments, there is a lack of emphasis on developing sarcasm detection models that prioritize energy efficiency [51, 52].
For this study, we consider the following research questions:

- How can depthwise convolutions be optimized to reduce the memory footprint of CNN while maintaining or reducing the energy consumption of CNN model on CPU-based systems?
- How does the energy efficiency of depthwise convolutional models vary across dif-

ferent datasets, and what insights can be gained to optimize their performance for specific use cases, such as audio sarcasm detection, while maintaining energy efficiency on CPU-based devices?

- How does the utilization of depthwise convolution impact the efficacy and performance of varying activation functions and pooling layers in neural network architectures?

The remainder of this research paper is organized as follows: In Section II, we review the existing

works of literature. Section III discusses the problem statement, and Section IV discusses our dataset creation. Section V discusses the proposed system, and Section VI discusses the results and experiments.

2 RELATED WORK

2.1 Sarcasm detection

In the current state-of-the-art, automatic sarcasm detection can be broadly categorised into text-based approaches, image-based approaches [53], and audio-based approaches. For the first approach, researchers have used algorithms like support vector machines (SVM), naive bayes, and deep learning like CNN and Long short-term memory (LSTM) to detect sarcasm from tweets and news headlines. The recent work published in 2024
[54] proposed a hybrid deep learning approach for sarcasm detection in Arabic, lever- aging fine-tuning three pre-trained transformer-based language models (LM). Another work [55] proposed a GRU-CAP for product feature sarcasm analysis based on online reviews. Another work [56] introduced the Quantum Fuzzy Neural Network (QFNN) for sentiment and sarcasm detection in social media. However, the current state of the art necessitates that users communicate through digital mediums, which is hence not suitable for real-time detection.

In the second approach, researchers have used emojis, images encapsulated with text. In both scenarios, sarcasm is identified through the concept of contradiction between the text context and the image-occurrence context. Some of the recent works

[57] proposed gate mechanisms and guide attention for image-text sarcasm iden- tification. Another work [58] proposed BERT and ResNet to detect sarcasm from text-image.Another work [59] proposed a bi-directional GRU to detect sarcasm using emoji. However, the current state of the art, which predominantly employs images for sarcasm detection, requires users to supplement these images with text. Therefore, these approaches are solely applicable to digital communication.
In the third approach, researchers have utilised audio combined with text and video combined with text to identify sarcasm utilising audio modality. The work [45] pro- posed a hierarchical attention module; they converted audio to text for detection. Another work [60]proposed combining mBART with GCN for detecting sarcasm from text converted from audio.Another work [14] included audio as one of the inputs com- bined with video and text to identify sarcasm. Another work [61] combined ViFi-CLIP and Wav2vec2 for BART for sarcasm identification from audio-video-text. While these solutions can be utilised in real-time contexts, their operation relies heavily on mul- tiple modalities of data for decision-making, leading to the need for multiple models for preprocessing and training. Making them incompatible with resource- and power- constrained devices.

And to solve this issue of data overloading, researchers have utilised unimodal solu- tions for sarcasm detection [62, 63]. Table 1 summarises the unimodal works from 2011–2024, chronologically arranged, gathered from SCI-indexed journals of ACM, IEEE, Elsevier, and Springer. Table 1 shows that there is a lack of audio-based uni-modal sarcasm detection applications that are energy-efficient and compatible with resource-constrained devices.

2.2 Existing multiply-and-accumulate reduction strategies

Multiplying and accumulating is the most important operation in a deep learning algorithm [101]. However, this operation requires huge resources. Which includes the expensive memory and CPU time [102]. As the number of operations increases, so does the consumption of resources [103, 104]. In the current state-of-the-art, MAC reduction techniques can be broadly classified into hardware-based and software-based approaches. For the first approach, researchers have utilised architectures like Non-von Neumann, the design of specialised accelerators, and approximate MAC unit design. Here we discuss some of the recent works. The work [105] proposed twofold sparsity,

Table 1 Comparison of Unimodal-Sarcasm Detection Algorithms from the year 2011-2024. Here [U] represents considered input, and [A] represents accuracy above 80%.

S.No.	[U]	Algorithm	[A]	Dataset
[64]	Text	Naive Bayes classifier	Yes	IAC
[65]	Text	Naive Bayes+SVM+Maximum Entropy	Yes	Amazon, Twitter, Sina Weibo
[57]	Text	Gate mechanism+Guide attention	Yes	Twitter
[64]	Text	Naive Bayes classifier	No	IAC
[66]	Text	SVM	Yes	Twitter
[67]	Text	Na¨ıve Bayes + fuzzy clustering	Yes	Twitter
[68]	Text	Naive Bayes classifier+Vader classifier	No	STSG, SSTb
[69]	Text	MLP+Guassian Naive Bayes	Yes	Twitter
[70]	Text	Context Aware CNN	Yes	Twitter
[71]	Text	LSTM	Yes	IACv2, Reddit, Twitter
[72]	Image	CNN+fuzzy inference system	No	COGNIMUSE
[73]	Text	CNN	Yes	Twitter
[74]	Text	CNN+Local Max pooling	Yes	Twitter, IAC
[75]	Text	SVM	Yes	Twitter
[76]	Text	BiLSTM+Attention layer	Yes	IAC
[77]	Text	Softmax attention layer BiLSTM	Yes	Twitter
[78]	Text	RoBERTa+BiLSTM	Yes	SemEval-2018
[46]	Text	Hierarchical Attention Network	Yes	MaSaC
[79]	Text	Pre-trained COMET model	No	Twitter, Reddit
[80]	Image	incongruity-aware attention network (IWAN)	Yes	MUStARD
[81]	Text	Google BERT	No	Twitter
[82]	Text	Bi-LSTM+GloVe	Yes	IAC-v2, Twitter
[83]	Text	ViT+BERT+GCN	Yes	Twitter
[84]	Text	GRU+ ELMo	No	IMDb, SSTb, SemEval, STSG
[85]	Text	Pre-trained AlBERToIS	Yes	IronITA
[86]	Image	Context Aware CNN	Yes	Twitter
[87]	Text	BiGRU+Attention+convolution	Yes	Twitter
[88]	Text	SVM+LSTM	Yes	Sarc-H
[89]	Image	Bi-GRU+Attention module	Yes	GuanSarcasm
[13]	Text	SVM+PSO	Yes	News headlines dataset
[86]	Text	Dual Channel CNN	Yes	Twitter, Reddit
[90]	Text	BiLSTM+Attention layer	Yes	IAC
[58]	Image	BERT	Yes	Twitter and Reddit
[91]	Text	CNN+LSTM	Yes	Sarc-H
[92]	Text	BiLSTM+GCN	Yes	Twitter, Reddit, IAC
[93]	Text	ViT+BERT	Yes	IMDB
[94]	Text	BERT+Attention module	Yes	IAC
[95]	Text	GT-BiCNet+ALABert	Yes	Twitter
[96]	Text	CNN+BiLSTM+BERT	Yes	Twitter, Reddit
[97]	Text	Encode+LSTM+CNN	No	MELD, MEISD, and MSED
[98]	Image	Mobilenetv3	Yes	Webscraped data from web
[99]	Text	LSTM+BERT	Yes	Twitter

[54]	Text	BERT+DNN	Yes	ArSarcasm
[100]	Text	Transformer encoder+Gating Network	No	MUStARD, Memotion
[?]	Text	Bipolar semantic attention	No	Twitter

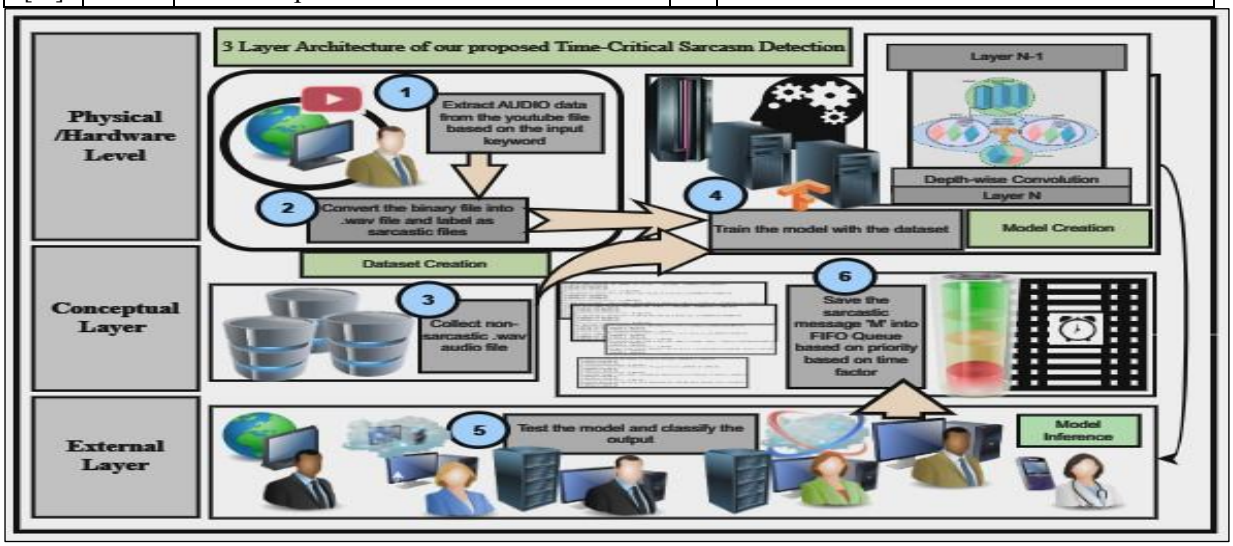

Fig. 1 Time-Critical Sarcasm Detection Architecture

a technique aimed at sparsifying deep learning models for energy-efficient computa- tions on edge devices [106]utilising compute-in-memory (CIM) architecture. Through joint bit- and network-level sparsity during training and the use of Linear Feedback Shift Register (LFSR) masks. Another work [107] introduces mem-resonators, com- bining metal-oxide memristors with microring resonators, for in-memory photonic computing. Another work [108] proposed a CANET framework for deploying quan- tized neural networks on resource-constrained platforms, enabling MAC operations with 8-bit accumulators. Another work [109] proposed an IMC macro based on cus- tom 9T-SRAM.Another work [110] proposed a recurrent neural network accelerator optimised for low-power edge sensor nodes. It utilises 8-bit quantization and hard sigmoid activation to reduce memory and resource requirements. However, these tech- niques are hardware-based and come with a series of issues with upgrading, and most of the solutions are tailored for a single type of application and do not support multi- domain. In the second approach, researchers have utilised pruning, quantization, and knowledge distillation and replaced convolution with less-cost operations. The [111] proposed framework includes server-side forward pass quantization and meta-learning, along with client-side backward pass quantization and gradient pruning. Another work [112] proposed a post-training quantization framework utilising coarse and fine weight splitting (CFWS). Another work [113] proposed a 1-D fully convolutional network architecture with depth-wise separable convolution (DSC) combined with atrous spa- tial pyramid pooling (ASPP) modules for efficient PPG artefact detection. Another work proposed [114] a federated learning framework leveraging spiking neural net- works for radar gesture recognition.However, the software-based works in this section are either CPU-on-edge device supported or energy- and performance-efficient, but not both.

2.3 Depthwise convolution

Depthwise convolution is a type of convolutional operation that, unlike the convolu- tion that operates on the entire input volume with a single filter, operates separately on each input channel, reducing the computational cost [115]. The depthwise con- volution operation is generally found in two different perspectives in the literature. The first one is a replacement, where the computationally intensive convolution is directly replaced with depthwise operations. And the

second one is optimising the depthwise convolution. In the first approach, researchers have considered design- ing lightweight deep learning solutions by replacing the existing convolutions with depthwise convolutions (DC). Work such as [116] proposed Dynamic Residual Con- volution (DRConv) by inserting 3×3 depthwise convolution between two 1×1 layers in the architecture. Another work [117] proposed a temporal depthwise convolutional transformer (TDCT). The model utilised DC with the self-attention mechanism of transformers. Another work [118] proposed multi-class cancer classification, integrat- ing depthwise convolutional networks with transformer architecture.Another work [119] proposed integrating depthwise separable convolutional blocks along with a sim- plified MobileNet V3 network. However, these models do not guarantee to reduce the parameters that fit a resource-constrained device.

In the second approach, researchers have optimised depthwise convolution opera- tions. Work [120] proposed a depthwise separable convolutional computing-in-memory (CIM) solution for edge AI applications by leveraging channel-wise parallel compu- tation. Another work [121] proposed finger vein recognition, utilising the residual pyramid-based depthwise separable MobileNetV3 (PyDS-MV3) technique to accu- rately identify finger veins. However, this work has not focused on parameter reduction in the layer. Another work [122] proposes RiSA, a DNN accelerator design aimed at enhancing PE utilisation for depthwise convolutions on systolic arrays through 1D PE chaining. However, the existing works in depthwise convolution have either focused on parameter reduction of the convolution operation or are being implemented as a hard- ware solution to accelerate the application through parallelization. There is a lack of research on reducing the MAC and parameters of depthwise convolution.

3 PROPOSED SYSTEM

3.1 Problem Statement

In the context of binary audio sarcasm classification, our goal was to create an energy-efficient model by optimising the architecture of a depthwise convolutional neu- ral network. This involved reducing parameters and minimising multiply-accumulate (MAC) operations to decrease energy consumption during training on a CPU-based system. The existing depthwise convolution layer is represented as:

Input volume: $W_{in} \times H_{in} \times D_{in}$ (width, height, depth)

Filter size: $F \times F$ (assuming square filter)

Stride: S

Padding: P

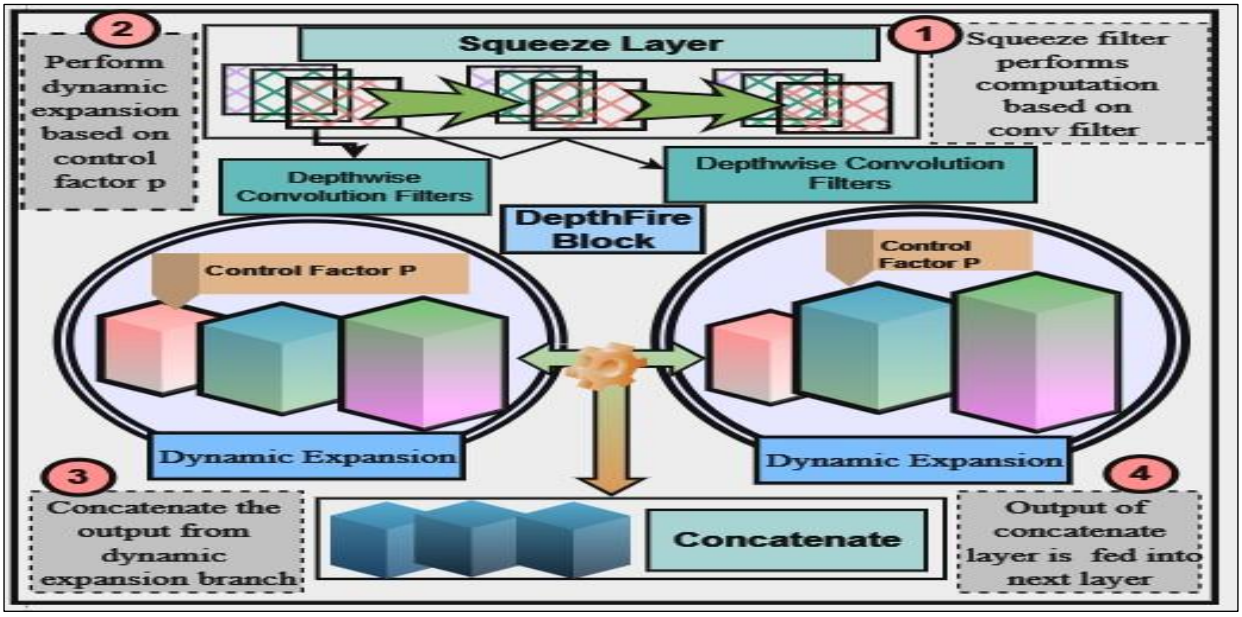

Fig. 2 Abstract representation of DepthFire Block for depthwise convolution parameter reduction

The output volume dimensions are calculated as follows:

$$\text{Output size: } W_{out} = \frac{'D_{in} + 2P - F'}{S} + 1 \quad (1)$$

Output size: W

$$\text{Depth: } D_{out} = D_{in}$$

The number of parameters in a depth wise convolutional layer is given by:

$$\theta_d w : F \times D_{in} \times 1 \quad (2)$$

Let T represent the total time duration of the audio signals in the dataset, MAC(t) denote the MAC operations at time t, and θ(t) signify the number of parameters in the model at time t. We formulate the problem as follows:

$$E(\theta) = \int_0^T MAC(\theta_d w, t)\, dt \quad (3)$$

And our goal is to Minimize:E(θ) for the task by reducing the parameters of the depthwise convolutional layer.

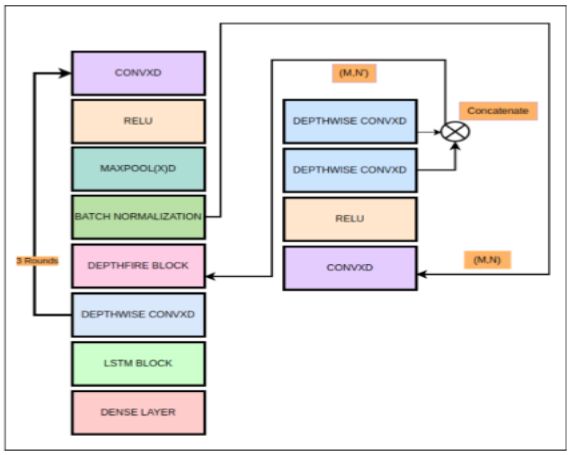

Fig. 3 Network architecture with DepthFire for binary sarcastic audio classifier with single epoch

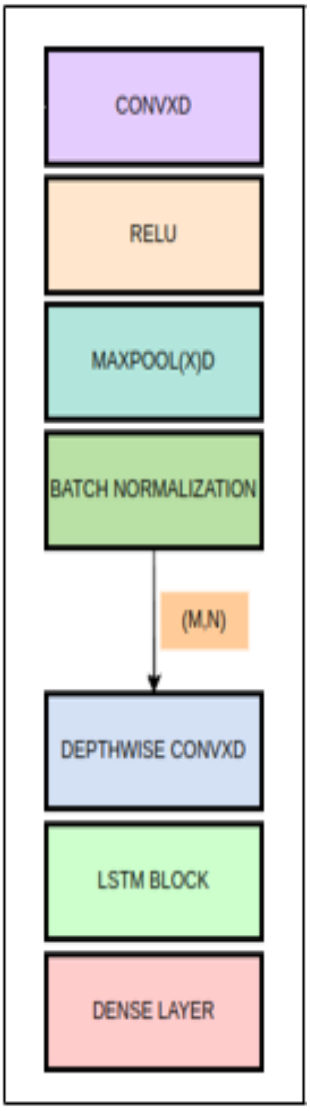

Fig. 4 Network archi- tecture without Depth-Fire block

3.2 Dataset Creation

As of best of our knowledge, there is no specific dataset available for audio only classification. Therefore, we gathered datasets from online sources.We extracted audio files from YouTube videos labeled as sarcastic. The algorithm used for this task is detailed below:

Algorithm 1: YouTube Audio Downloader

```
Data: topic
Result: Download or extract audio files from YouTube links
scrape youtube links(topic);
foreach link in YouTube links do
    Extract Audio from YouTube video using youtube dl;
    Save the downloaded audio file;
end
```

Here we considered the following web series found on YouTube, which are labelled as sarcastic. The series is as follows: Friends, Bing Bang Theory (2007), and Sarcamo- holics. And then we extracted the audio files as wav-type files. Here, we succeeded in collecting 101 files from the web after multiple loops. For training the model for
non-sarcastic audio identification, we considered the CREMA-D dataset[123]. To the best of our understanding,our analysis of the CREMA-D dataset did not uncover any audio files containing sarcasm. We examined the presence of words ending with long syllables [7] and could find no instances of sarcasm.

3.3 Data-preprocessing and training

Supervised learning was applied to develop our classification model using a dataset comprising 7,448 samples of sarcastic and non-sarcastic audio. The data was divided into a 75:25 ratio for training and testing purposes, respectively. Output labels were encoded using one-hot encoding, representing the two classes: sarcastic and non-sarcastic. To capture relevant speech features, we extracted MFCC, ZCR, spectral centroid, roll-off, bandwidth, and Mel-frequency cepstral delta coefficients using the Python-based Librosa library, following established methodologies from prior studies [124–126]. Subsequently, we implemented a CNN-based architecture (refer to Figure 3) for model training. Further details about the network architecture are provided in the subsequent discussion.

3.4 Proposed Architecture

60(

3.4.1 Time-Critical prediction of sarcasm

According to our study, we assessed that sarcasm is time-dependent, meaning that a sarcastic message 'M' changes its nature over time. A message deemed sarcastic at time 't' may not retain its sarcasm at time 't+1' due to a change in the state of mind of the speaker and listener. To address this, we introduced a time-critical sarcasm detection model. In this model, we incorporated a priority queue where messages were assigned higher priority based on their earliest timestamp. This approach was mathematically formulated as follows:

Message representation:

- M(t): Represents the message at time t.

Priority degradation:

- Priority(M(t), t): Defines the priority level assigned to message M(t) at time t.
- Priority(M(t), t) decreases over time t.

Priority in this case is represented as :

$$\text{Priority}(M(t), t) = f(t) \quad (4)$$

Where f(t): A function representing the degradation of message priority over time.

Capability to convey sarcasm:

- Capability(M(t), t): Represents the capability of the model to capture message M(t) that convey sarcasm at time t.
- The capability to capture sarcasm, Capability(M(t), t), is influenced by the priority level of the message.

And capability to capture sarcasm by the message M is represented as:

$$\text{Capability}(M(t), t) = g(\text{Priority}(M(t), t)) \quad (5)$$

Where g(·): A function mapping the priority level of the message to its capability to convey sarcasm.

The goal was to enhance the deep learning model's ability to detect sarcastic mes- sages, prioritising those with higher urgency levels. Therefore, we defined the objective function as follows:

$$\text{Maximize Capability}(M(t), t) \quad (6)$$

Following classification, a FIFO queue was

utilised to retain the data for future reference.

Algorithm 2: Time-critical allocation of sarcastic message to FIFO queue

```
Input: Q[n]: Size of FIFO queue, rear
Input: sarcasm_audio: Sarcastic audio message to be enqueued
if *rear = size of Q − 1 then
    Enqueue sarcasm_audio into Q with highest priority;
    Q ← [sarcasm_audio[q1, q2, . . . , qn]];
    Decrease the priority of existing messages in Q;
    for i from 1 to n do
        priority(qi) ← priority(qi) − 1;
        if priority(qi) ≤ 0 then
            Dequeue the message at index qi from Q;
        end
    end
end
else
    Dequeue the oldest message (message with least priority) from Q;
end
```

The detailed algorithm outlining this process is provided below. The architecture of the binary audio sarcasm detection system, employing the proposed CNN model, is depicted in Figure 1. Algorithm 3 outlines the procedure for time-based sarcasm prediction.

Algorithm 3: DepthFire Block for Depthwise Convolution parameter reduction

```
Data: x, S, p, state
Result: Output tensor representing the concatenation of expanded features
procedure FIREBLOCK(x, S, p):

if state == 'Training' then
    squeezed ← ConvXD(x, S) ;
    p_threshold ← Randomly sample from U(0, 1) ;
    if p < p_threshold then
        expanded_1x1 ← DepthwiseConvXD(squeezed);
        expanded_3x3 ← DepthwiseConvXD(squeezed);
    else
        expanded_1x1 ← x                    ▷ Skip the layer;
        expanded_3x3 ← x                    ▷ Skip the layer;
    end
else
    expanded_1x1 ← DepthwiseConvXD(squeezed);
    expanded_3x3 ← DepthwiseConvXD(squeezed);
end
output ← concatenate([expanded_1x1, expanded_3x3])
return output
end procedure
```

3.4.2 Proposed approach for depthwise convolution parameter reduction

In this study, we introduced the DepthFire block to reduce the parameters associated with depthwise convolutions within the network. Our inspiration stemmed from the Squeezenet architecture, which initially focused on reducing input dimensionality using fire-and-excite blocks. The original Squeezenet model applied parameter reduction to the entire CNN but did not specifically address parameter reduction for depthwise convolutions. To address this gap, we proposed the DepthFire block to precede the depthwise convolution, thereby reducing the complexity of both the

depthwise con- volution and the overall CNN model. Our approach involved dynamically adjusting the number of depthwise convolutions based on a control factor P, enabling flexi- ble network complexity adjustments. Figure 1 illustrates a three-layer architecture incorporating the proposed DepthFire block. DepthFire block As shown in Fig.3, the proposed model performs a dual depthwise convolution. Where the depth of the depth- wise convolution is controlled by a control factor P to perform computation without increasing the computation burden. Then two depthwise convolutions are concate- nated, and the final output is passed to the depthwise layer in the network. Algorithm 3 illustrates the entire process of parameter reduction of the depthwise convolution

process in Fig.2. Let $\mathbf{I} \in \mathbf{R}^{H \times W \times Ci}$ denote an input feature map produced by $\mathbf{L}_{n-1}$,
and (M, N) denote the output from the layer L_{n-1}. This layer is assumed to be placed just before the proposed block. We feed (M, N) into the" depthfire" block, which con- sists of a convolutional layer with X dimension followed by a depthwise convolution with X dimension, as shown in Fig.2. Here, M denotes the output size from the layer, and N denotes the number of channels or feature maps in the output. For each layer, we compute M as follows:

$$M = \frac{\text{Input Size} - \text{Kernel Size} + 2 \times \text{Padding}}{\text{Stride}} + 1 \quad (7)$$

And N denotes the number of filters and estimated number of feature map.And the number of parameters in each layer is computed as:

Number of Parameters = kernel Size × input channel+ output channel × Num filters (8)
In contrast to the depicted scenario in Fig.4, where the output from the preceding layer of size (M,N) is directly fed into the depthwise convolution, we modified the size of N according to the principle of the depthwise convolution layer resulting in N′. In this configuration, each filter operates independently on each input channel, ensuring that the input and output channels are of equal size. In this configuration, the feature maps produced by the depthwise layers are concatenated along the channel dimension to form the final output feature map for the depthwise convolution layer within the network. The depthwise convolution operation preserves the spatial dimensions H×W
of the input feature map. Within the "depthfire" block, each dimension X contains
two separate branches of depthwise convolution layers, each with a X × X kernel size controlled by a factor P. A higher value of P leads to a reduction in parameters. Overall, our "depthfire" block effectively reduces the number of parameters of the deep convolutional layer within the network. Furthermore, as shown above, this leads to a reduction in the number of MAC operations since there are fewer input channels for the subsequent layer.

4 EXPERIMENTS AND RESULTS

To Validate the proposed lightweight-based sarcasm detection we conducted various tests. And furthermore, to validate our proposed method depthFire, we set up an experimental platform and conducted different tests using different datasets. Our net- work model was compared against others to assess its effectiveness. Additionally, we studied the impact of depthwise convolution size on the pooling layer and activation functions to verify classification accuracy and compression effects using a control fac- tor P. These experiments were performed on a computer equipped with an AMD Ryzen CPU running at 1965.9 MHz and a Radeon graphics card, as well as on the Google Cloud CPU and Amazon Sagemaker CPU platforms. Energy consumption was measured using pyRAPL, a software toolkit that estimates power consumption dur- ing Python code execution, utilising Intel's 'Running Average Power Limit' (RAPL) technology. The following sub-sections discuss the experiments conducted in detail.

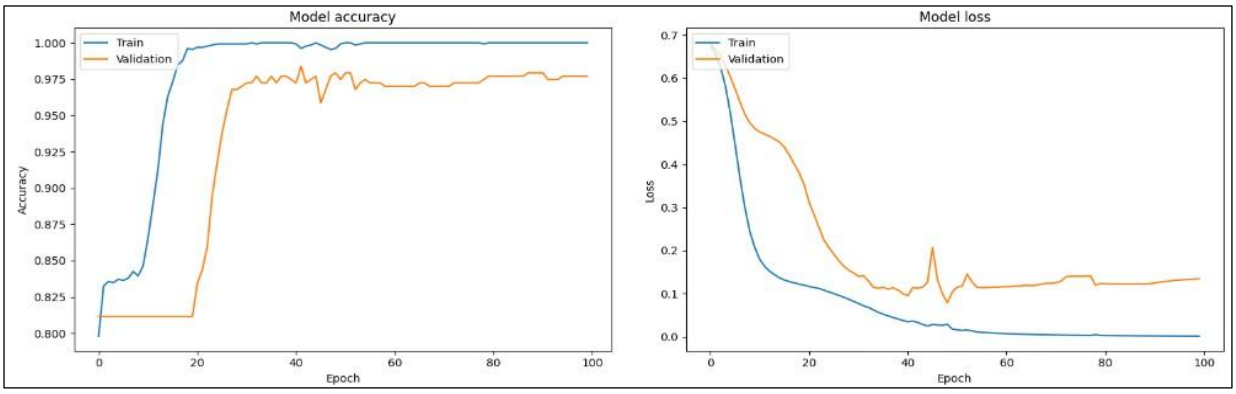

Fig. 5 Accuracy and loss for Train-Test Sklearn library with 75:25 ratio

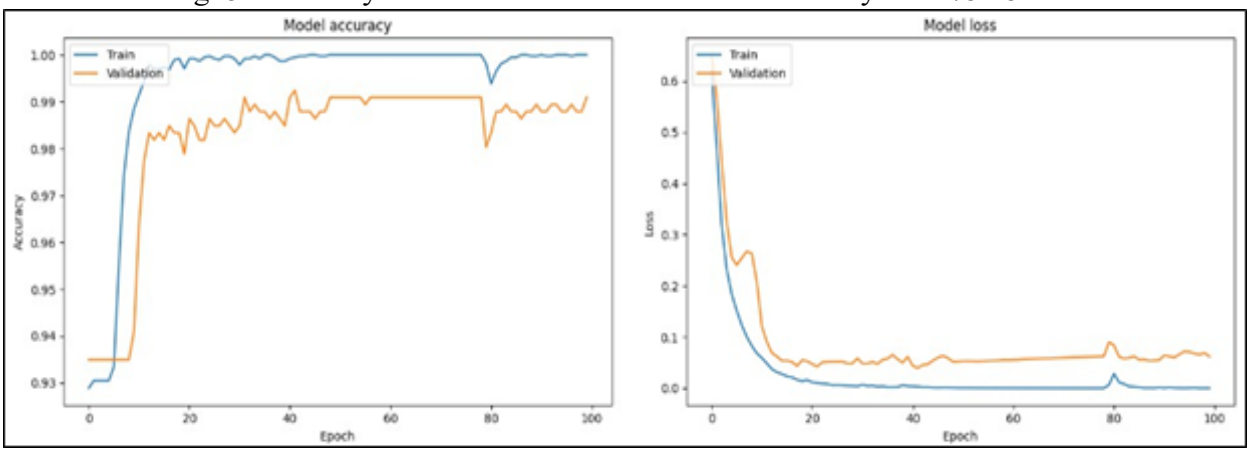

Fig. 6 Accuracy and loss for Train-Test Sklearn library with 50:50 ratio

4.1 Sarcastic audio classifier

To showcase the performance of our proposed model, as depicted in Fig. 3, we conducted a comparison with other networks and demonstrated their classification accuracy using a custom-created YouTube fetched dataset. We compared the proposed model with CNN [127], CNN+LSTM [128], CNN+L1, CNN+L2, and CNN+Dropout [129].Table 4 illustrates that our proposed network model exhibits higher accuracy with a lesser number of parameters on the custom dataset. Moreover, our model achieves this high accuracy while reducing the number of network parameters relative to these other models. To validate our study, we performed Validation using Sklearn library and K-cross Validation. Figures 5,7,6, 9, 8 illustrate the accuracy and loss related to training and validation respectively.And Figure 10 and 11 show the confu- sion matrix related to the model for train and test ratios 50:50 and 75:25 ratio.

Table 2 Comparison of Sarcasm Detection Methods with the proposed method

Method	Accuracy	Parameter	User Time	System	Wall Time
LSTM+MFCC [130]	0.3763	384.96 KB	4min 45s	18.1 s	5min 21s
Xception+MFCC [131]	0.9996	6.01 MB	38min 6s	9min 18s	53min 26s
LSTM+RNN+MFCC [132]	0.9989	1.34 MB	5min 7s	2min 18s	9min 56s
CNN+Dropout [129]	0.796	53.97 MB	24.6 s	2.34 s	26.9 s
CNN+LSTM [127]	0.7929	172.94 KB	1h 5min 47s	12min 23s	4min 9s
CNN+L2+MFCC+ZCR	0.99	53.97 MB	19h 22min 3s	2h 59min 45s	6h 5min 42s
CNN+L1+MFCC+ZCR	0.926	53.97 MB	19h 33min 45s	3h 6min 46s	6h 20min
Proposed	99.8586	185.95 KB	40min 53s	30min 55s	1min 42s

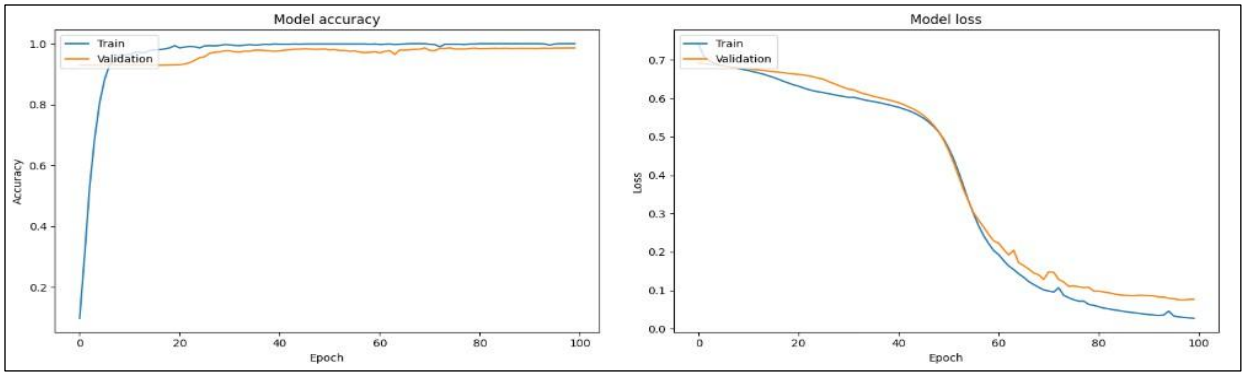

Fig. 7 Accuracy and loss for Train-Test Sklearn library with 25:75 ratio

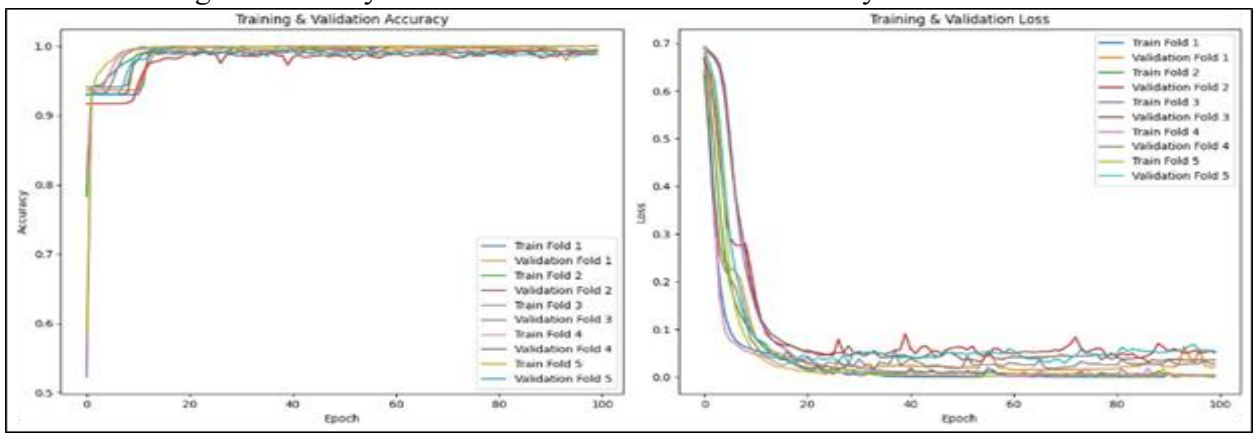

Fig. 8 Accuracy and loss with K-cross validation with K-split= 5

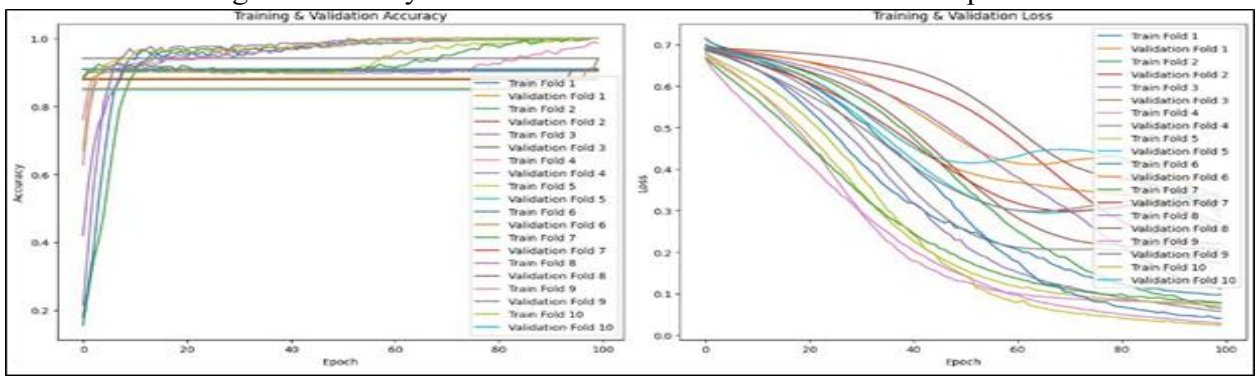

Fig. 9 Accuracy and loss with K-cross validation with K-split= 10

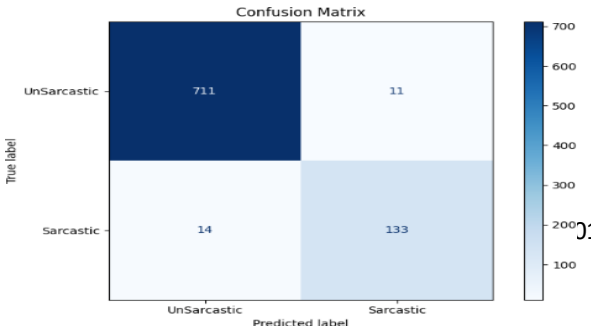

Fig. 10 Confusion Matrix representation of the proposed model on 50:50 ratio

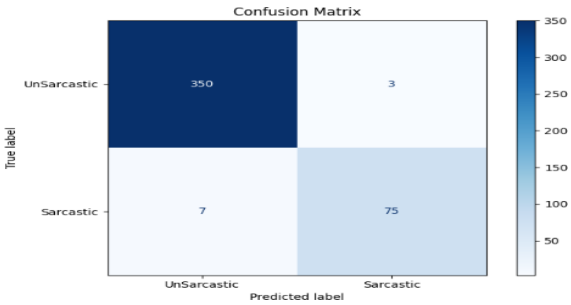

Fig. 11 Confusion Matrix representation of the proposed model on 75:25 ratio

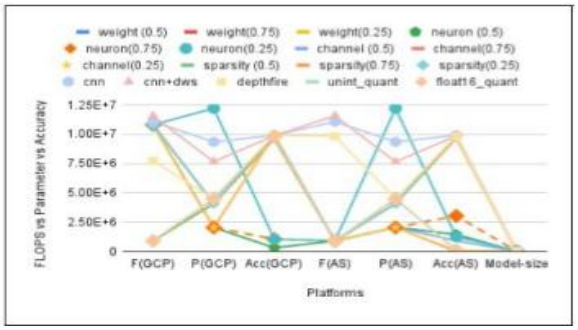

Fig. 12 Comparative Analysis of Accuracy represented as Acc, FLOPs represented as F, Parameter Sizes represented as P and Model size for Pruning and Quantization Techniques on the MNIST dataset implemented on Google Cloud Platform(GCP) and Amazon SageMaker Cloud(AS)

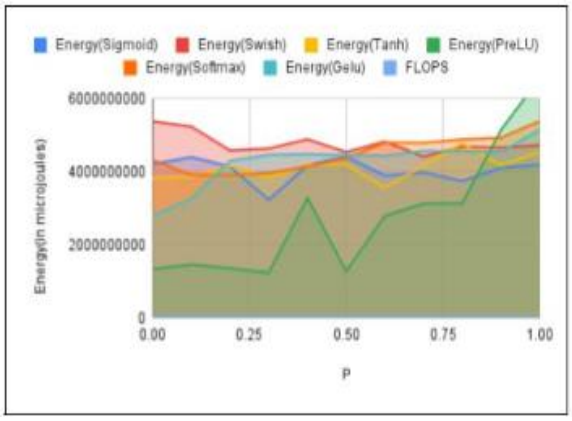

Fig. 13 Comparison of the energy consumption and FLOPs on proposed Depthfire with different activation functions on MNIST dataset

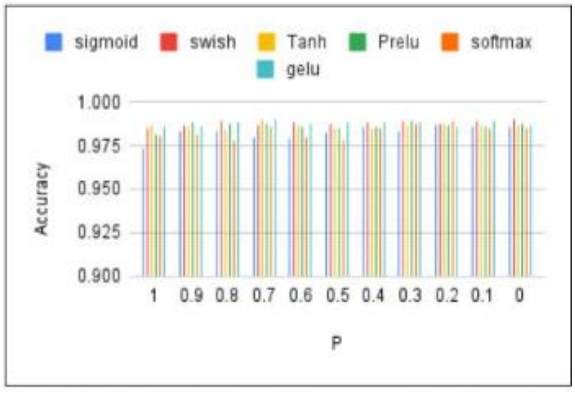

Fig. 14 Comparison of the accuracy on proposed Depthfire with varying activation func- tion on MNIST dataset

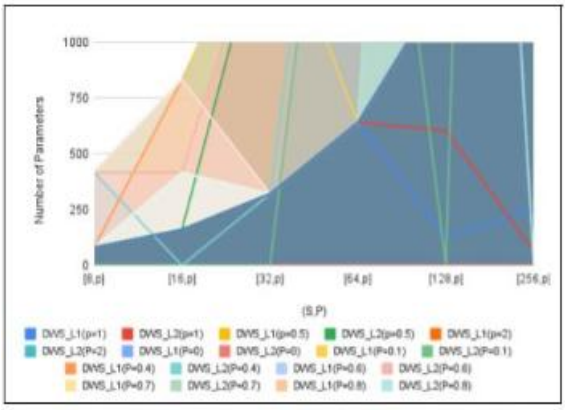

Fig. 15 Comparison of Depthwise parameters with different control factor P on MNIST dataset

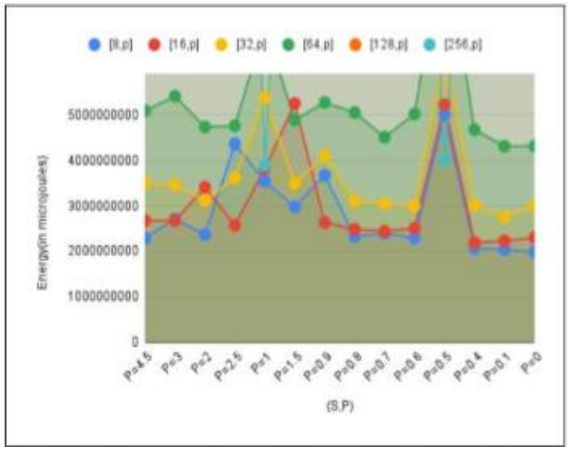

Fig. 16 Comparison study of energy-consumed with MNIST dataset on different control factor P

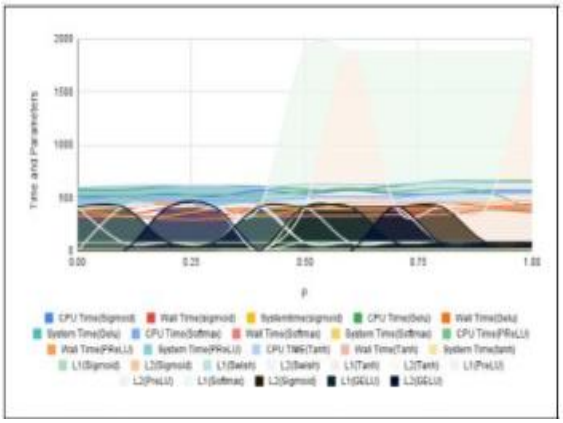

Fig. 17 Comparison study of CPU, System and wall time, depthwise convolution parameters with CIFAR-10 dataset on different control factor P

4.2 Pruning and quantization algorithms

To assess our model's computational requirements and compare them with alternative techniques, we conducted a benchmarking study focusing on floating-point operations (FLOPs) and multiply-and-accumulate computation (MAC). This evaluation involved pruning and quantization techniques applied to our model on the MNIST dataset, tested on the Google Cloud CPU and Amazon SageMaker CPU platforms to test it adaptability on different platforms. We compared various pruning methods, including channel pruning, neuron pruning, magnitude-based weight pruning, sparsity of filter pruning, and float16 model weight quantization.

For the experiments, we used the network architecture depicted in Fig. 3, adapting it for MNIST by removing the LSTM layer due to the absence of time-varying inputs in this dataset. Figure 12 presents the results of these experiments conducted on MNIST. Our findings indicated that the model size remains independent of FLOPs, and metrics such as system time, CPU time, and wall time vary with changes in model size and parameters for a constant FLOP count. Notably, the model and parameter sizes were consistent across both cloud platforms, with similar FLOPs observed. From this study, we conclude that in deep learning, the FLOP count is not directly correlated with execution time or model size. Figure 12 illustrates these metrics, where accuracy is scaled to 10^7 and parameters to 10^2.

4.3 Different control factors p and activation functions

This study examined the influence of the control factor P on squeeze filters to adjust the depth of the depthwise convolution layer, impacting parameters, classification accuracy, model size, and energy consumption. Previous experiments demonstrated the effectiveness of our model with the Rectified Linear Unit (ReLU) activation function. To further investigate its versatility with different nonlinearities regarding energy consumption and model size, we compared network performance across vari- ous P values, squeeze filter sizes, and activation functions using the MNIST datasets. These experiments were performed on an AMD Ryzen 5 CPU using PyRAPL.

4.3.1 Different activation functions

Here we tested on MNIST dataset.Figures 14 and 13 demonstrate the changes in classification accuracy and energy consumption across various activation func- tions, including Rectified Linear Unit (RELU), Gaussian Error Linear Unit (GELU), Parametric Rectified Linear Unit (PRELU), Softmax Function, Hyperbolic Tangent, Sigmoid, and Swish. Notably, at P = 0.5, PRELU, Softmax, and GELU exhibited a substantial increase in energy usage, whereas Swish and Sigmoid displayed an oppo- site trend. Swish showed the highest energy consumption at P = 0, contrasting with the minimal usage observed with PRELU. Sigmoid exhibited maximum energy con-sumption at P = 0.1, whereas PRELU demonstrated the lowest.

Throughout various P values, PRELU consistently maintained minimal energy usage, while other activation functions exhibited fluctuating patterns. For instance, at P = 0.2, PRELU's energy consumption was minimal, while Sigmoid reached its peak.

Similarly, Softmax recorded the highest consumption at P = 0.8, contrasting with PRELU's minimal usage.Despite variations in energy usage among different activation functions and control factors P, the model's accuracy remained robust and consis- tently high across all P values. Additionally, the floating-point operations per second (FLOPS) decreased notably after reaching P = 0.2. Interestingly, regardless of changes in the activation function, the model exhibited similar patterns in FLOPS reduction, indicating that FLOPS are independent of the type of activation function used.In con- clusion, decreasing the control factor P significantly impacted both the model size and FLOPS, with notable reductions observed beyond P = 0.3. Furthermore, the type of activation function did not significantly influence the observed patterns in FLOPS reduction.

Table 3 Comparison of parameters and trainable parameters on Deepfake application with the proposed system

Model	CPU Time	Wall Time	System Time	Size(MB)	Parameters	Trainable	Non-Trainable	Energy	Accuracy
Efficient Net B0	1min 11s	36.5s	4.38s	15.45	4,049,564	4,007,548	42,016	688,382,629	82.75
Efficient Net B1	1min 43s	57.9s	5.01s	25.08	6,575,239	6,513,184	62,055	964,852,553	75
Efficient Net B2	1min 45s	58.9s	3.32s	29.63	7,768,569	7,700,994	67,575	990,106,344	93.75
Efficient Net B3	2min 16s	1min 7s	15.9s	41.14	10,783,535	10,696,232	87,303	1,239,961,800	75
Efficient Net B4	3min	1min 16s	21.1s	67.42	17,673,823	17,548,616	125,207	1,540,665,732	56.25
Mobilenet	38.4s	20.7s	1.81s	12.32	3,228,864	3,206,976	21,888	410,928,578	87.5
MobilenetV2	45.6s	23.3s	3.76s	8.61	2,257,984	2,223,872	34,112	480,443,660	93.75
MobilenetV3small	22s	18.8s	1.68s	3.58	939,120	927,008	12,112	363,737,507	81.25
DepthFire	21.5s	4.9s	924ms	1.47	384,469	384,469	0	122,099,811	50
Xception	2min 2s	43.6s	9.21s	79.58	20,861,480	20,806,952	54,528	899,581,775	87.5
VGG16	2min 45s	33s	9.66s	56.13	14,714,688	14,714,688	0	880,246,623	50
VGG19	3min 5s	37s	10s	76.39	20,024,384	20,024,384	0	891,479,243	43.75
RESNET50	2min 19s	42.2s	4.9s	89.98	23,587,712	23,534,592	53,120	884,315,016	50
REsnet101	3min 33s	1min 42s	10.2s	162.73	42,658,176	42,552,832	105,344	1,604,059,930	62.5
NASnetMobile	2min 1s	1min 35s	3.82s	16.29	4,269,716	4,232,978	36,738	1,577,686,477	87.5

4.3.2 Different squeeze filter

Here we experimented on the MNIST dataset.Figures 15 and 16 present the results of our study, where we evaluated different squeeze filter configurations (8, 16, 32, 64, 128, and 256). Notably, the configuration with 8 filters exhibited the lowest energy consumption, while the configuration with 256 filters consistently showed the highest energy consumption across all P values. Specifically, at P = 0.5, the energy consump- tion was maximised, whereas it was minimised at P = 0, P = 0.1, and P = 0.4.Figure 16 highlights that the depthwise convolution layer demonstrated the lowest energy consumption when P = 1. Additionally, there was a notable decrease in the number of parameters in the depthwise convolution layer as P increased from 0.6 to 1.

In summary, as P approached 1, the number of parameters in the depthwise convolution layer decreased, but the energy consumption increased.

4.3.3 Testing with CIFAR10

The results obtained from previous sections demonstrated good performance of the proposed network on the MNIST dataset. To explore the transferability of our model, we conducted an analysis on the CIFAR-10 dataset, as depicted in Fig. 17. We com- pared the depthwise convolution parameters with CPU, system, and wall time to

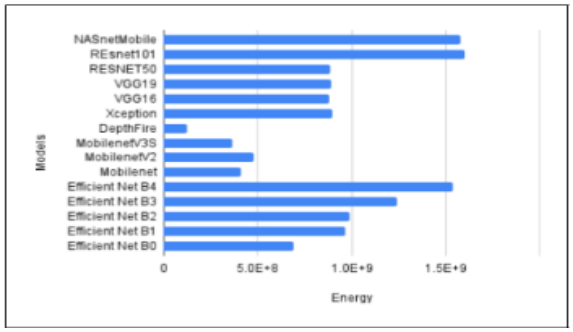

Fig. 18 Comparison of energy consumption for Deepfake detection application with the proposed system

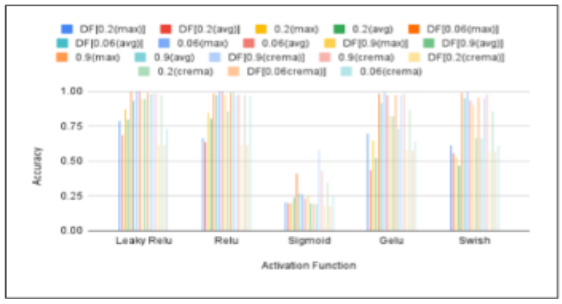

Fig. 19 Comparison of accuracy across different neural network Configurations with Max, Average Pooling and different activation functions on RAVDESS and CREMA-D with and without DepthFire

assess the adaptability of our model with larger datasets in terms of time efficiency. Specifically, CPU time represents the duration during which the central processing unit (CPU) executes instructions for a given task, reflecting the actual computational workload. System time, also known as kernel time, encompasses the period in which the CPU executes kernel-level instructions on behalf of the operating system, includ- ing system calls and interrupts. Wall time, or real-time elapsed time, indicates the total duration from the start to the completion of a task.

In our study, we observed that the sigmoid activation function exhibited the lowest CPU, wall, and system times at $P = 0$, with these values increasing as P increased. Conversely, Swish demonstrated the lowest CPU time at $P = 0.5$, and a comparable minimum was observed in wall time at the same P, while the system time minimum occurred at $P = 0.1$. For Softmax, the CPU time was minimized at $P = 0.2$ and $P = 0.3$, while the system time reached its minimum at $P = 0.3$. PRELU showed the lowest CPU time at $P = 1.0$, with the minimum wall time observed at $P = 0.2$ and the system time at $P = 0.3$. The Tanh activation function exhibited a pattern similar to Sigmoid in terms of time efficiency.

4.4 Result on other binary classification task

4.4.1 Binary Video classification-Deepfake detection

To assess the adaptability of our model to a different binary task within a distinct domain, we focused on deep-fake detection, a critical societal concern due to the escalating risks associated with deep-fake technology. We utilised the deepfake dataset from Kaggle for this study, conducting experiments on the AMD Ryzen CPU. We evaluated 18 pre-trained models, as detailed in Table 3. Our findings demonstrated that our proposed system surpassed others in terms of energy efficiency and model size, as depicted in Figure 18.

Table 4 Evaluation of DepthFire on Brain Tumour Dataset

Model	Parameters	CPU Time	Wall Time	System Time	Model Size	Accuracy	Trainable	Non-Trainable	FLOPS (GB)
CNN	4619524	1h 42min 1s	1h 17min 11s	10min 42s 60:	17.62 MB	0.9445	4619524	0	1.02
DepthFire	1110772	42min 57s	33min 26s	9min 10s	4.24 MB	0.9269	1110772	0	0.664
EfficientNetB0	9293735	1h 37min	1h 6min	3min 6s	35.45 MB	0.9498	9251712	42023	0.268

		26s	21s						
EfficientNetB1	11819403	2h 22min 46s	1h 32min 44s	3min 30s	45.09 MB	0.9522	11757348	62055	0.3914
EfficientNetB2	13537021	43min 24s	30min 13s	3min 29s	51.64 MB	0.9638	13469446	67575	0.451
EfficientNetB3	17076275	58min 45s	42min 16s	6min 52s	65.14 MB	0.9458	16988972	87303	0.6541
EfficientNetB4	25015139	1h 18min 36s	57min 41s	13min 45s	95.43 MB	0.949	24889932	125207	1.014
EfficientNetB5	36903419	1h 57min 26s	1h 20min 51s	9min 17s	140.78 MB	0.9541	36730676	172743	1.5822
EfficientNetB6	50398611	3min 34s	4min 46s	32 s	192.26 MB	0.9471	50174172	224439	2.24
EfficientNetV2B0	11163476	26min 53s	17min 46s	1min	42.59 MB	0.9393	11102868	60608	0.4839
EfficientNetV2B1	12175288	35min 39s	23min 30s	1min 43s	46.45 MB	0.9281	12104216	71072	0.67973
EfficientNetV2B2	14537826	40min 11s	26min 43s	1min 44s	55.46 MB	0.9272	14455538	82288	0.7814
SqueeezeNet	1236952	16min 35s	11min 33s	1min 8s	4.03 MB	0.6351	1236952	0	0.609

4.5 Result on multiclass classifier

Our experimental results have shown that the proposed solution is well suited for a binary classifier for performing energy-efficient CPU-on-device training. To assess the suitability of our proposed solution for multiclass classification tasks, we conducted tests in two different domains: audio and image.

4.5.1 Multiclass audio emotion detection

In this study, we explored the impact of filter size variation and different activation functions on the RAVDESS and CREMA-D datasets to analyse the effects of these parameter settings on the accuracy and energy consumption of CNNs. Adjusting the filter size influences the number of parameters, which in turn affects classification accuracy, model size, and energy usage.

Experiments were conducted on the AMD Ryzen 5 CPU using the TensorFlow frame- work, manipulating the filter size of convolutional filters to produce models with parameter counts of 0.9M, 0.2M, and 0.06M, as illustrated in the network architecture diagram (Fig. 3). The architecture details are further outlined in Table **??**, revealing that while the proposed DepthFire module with varying activation functions and a fixed kernel size maintained consistent model size and parameter counts across archi- tectures, there were notable variations in CPU time, wall time, system time, accuracy, and energy consumption.Results depicted in Fig. 19 and Fig. 20 suggest that within the same architecture, model size remains constant, with variations in accuracy and energy consumption attributed to differing operations performed. Consequently, we conclude from this section that activation functions influence execution time, accu- racy, and energy consumption in deep learning models independently of model size and parameter count.

4.5.2 Multiclass brain tumour detection

In this section, our proposed system was evaluated using an image-based multi-class classifier to evaluate the applicability of the proposed system on image data, specifi- cally using the Brain Tumour MRI Dataset sourced from Kaggle. The

implementation was carried out on the Tensorflow framework running on a Google Cloud CPU. The results and evaluation outcomes are presented in Table 4. The findings indicated that the proposed model demonstrated better performance compared to other models in terms of execution time and model size.

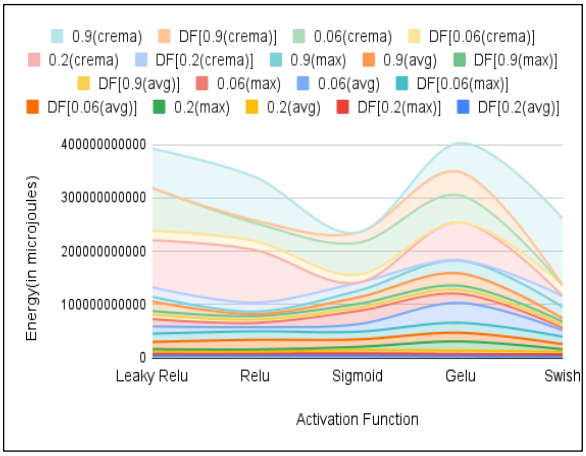

Fig. 20 Comparison of energy consumption between different neural network configurations on RAVDESS with max and average pooling and CREMA dataset on various activation functions com- pared with and without DepthFire

5 CONCLUSION

In this study, a lightweight and energy-efficient neural network model was proposed by integrating a depthfire block with depthwise convolution to reduce the computa- tional burden of CNN models, enabling deployment on resource-constrained devices. Experimental results demonstrated that the proposed model achieved a good balance between energy consumption and model size. The focus of this research was on reduc- ing depthwise convolution parameters through a case study on binary audio sarcasm detection. The proposed system was tested on AMD Ryzen CPU, Google Cloud CPU, and Amazon Cloud CPU across various datasets and applications. However, the eval- uation of this work was conducted in a controlled environment. As a future direction, we propose to assess the performance of the model in real-world scenarios, evaluating its failure rate and energy consumption over extended durations.

6 ACKNOWLEDGEMENTS

This research work was partially supported from GOOGLE Cloud Research Credits Program under EDU Credit 324579202.

7 DATASET AVAILABILITY

The dataset used in the research will be made available on request.

REFERENCES

[1] Kuchinad, K., Park, J.R., Han, D., Saha, S., Moore, R., Beach, M.C.: Which clin- ician responses to emotion are associated with more positive patient experiences of communication? Patient Education and Counseling 124, 108241 (2024)

[2] Demuro, E., Gurney, L.: Artificial intelligence and the ethnographic encounter: Transhuman language ontologies, or what it means “to write like a human, think like a machine”. Language & Communication 96, 1–12 (2024)

[3] Kotian, A.L., Nandipi, R., Ushag, M., Veena, G., *et al.*: A systematic review on human and computer interaction. In: 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), pp. 1214–1218 (2024). IEEE

[4] Mustafa, H.H., Darwish, N.R., Hefny, H.A.: Automatic speech emotion recogni- tion: a systematic literature review. International Journal of Speech Technology, 1–19 (2024)

[5] Chen, W., Lin, F., Li, G., Liu, B.: A survey of automatic sarcasm detection: Fun- damental theories, formulation, datasets, detection methods, and opportunities. Neurocomputing, 127428 (2024)

[6] Chouinard, B., Pesquita, A., Enns, J., Chapman, C.: Processing of visual social-communication cues during a social-perception of action task in autistic and non-autistic observers. Neuropsychologia, 108880 (2024)

[7] Cheang, H.S., Pell, M.D.: The sound of sarcasm. Speech communication 50(5), 366–381 (2008)

[8] Pexman, P.M., Olineck, K.M.: Does sarcasm always sting? investigating the impact of ironic insults and ironic compliments. Discourse Processes 33(3), 199– 217 (2002)

[9] Slugoski, B.R., Turnbull, W.: Cruel to be kind and kind to be cruel: Sarcasm, banter and social relations. Journal of Language and Social Psychology 7(2), 101–121 (1988)

[10] Ghosh, A., Veale, T.: Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 482–491 (2017)

[11] Creusere, M.A.: Theories of adults' understanding and use of irony and sarcasm: Applications to and evidence from research with children. Developmental Review 19(2), 213–262 (1999)

[12] Jose, J.M., Jeeva, J.: Energy-reduced bio-inspired 1d-cnn for audio emotion recognition. International Journal of Scientific Research in Computer Science, Engineering and Information Technology 11(3) (2023)

[13] Vinoth, D., Prabhavathy, P.: An intelligent machine learning-based sarcasm detection and classification model on social networks. The Journal of Supercom- puting 78(8), 10575–10594 (2022)

[14] Castro, S., Hazarika, D., P´erez-Rosas, V., Zimmermann, R., Mihalcea, R., Poria, S.: Towards multimodal sarcasm detection (an obviously perfect paper). arXiv preprint arXiv:1906.01815 (2019)

[15] Nath, K., Sarma, K.K.: Separation of overlapping audio signals: A review on current trends and evolving approaches. Signal Processing, 109487 (2024)

[16] Mughal, N., Mujtaba, G., Kumar, A., Daudpota, S.M.: Comparative analysis of deep natural networks and large language models for aspect-based sentiment analysis. IEEE Access (2024)

[17] Kar, T., Kanungo, P., Mohanty, S.N., Groppe, S., Groppe, J.: Video shot- boundary detection: issues, challenges and solutions. Artificial Intelligence Review 57(4), 104 (2024)

[18] Archana, R., Jeevaraj, P.E.: Deep learning models for digital image processing: a review. Artificial Intelligence Review 57(1), 11 (2024)

[19] Sponner, M., Waschneck, B., Kumar, A.: Adapting neural networks at runtime: Current trends in at-runtime optimizations for deep learning. ACM Computing Surveys (2024)

[20] Yang, Y., Wang, C., Gong, L., Wu, M., Zhou, X.: Conv-inheritance: A hardware-efficient method to compress convolutional neural networks for edge applications. Neurocomputing 487, 172–180 (2022)

[21] Kuo, C.-C.J., Madni, A.M.: Green learning: Introduction, examples and outlook. Journal of Visual Communication and Image Representation 90, 103685 (2023)

[22] Cheng, J., Wang, P.-s., Li, G., Hu, Q.-h., Lu, H.-q.: Recent advances in efficient computation of deep convolutional neural networks. Frontiers of Information Technology & Electronic Engineering 19, 64–77 (2018)

[23] Habib, G., Qureshi, S.: Optimization and acceleration of convolutional neural networks: A survey. Journal of King Saud University-Computer and Information Sciences 34(7), 4244–4268 (2022)

[24] Zhang, Q., Zhang, M., Chen, T., Sun, Z., Ma, Y., Yu, B.: Recent advances in convolutional neural network acceleration. Neurocomputing 323, 37–51 (2019)

[25] Zhao, B., Guo, J., Yang, C.: Understanding the performance of learning precod- ing policies with graph and convolutional neural networks. IEEE Transactions on Communications (2024)

[26] Mceliece, R.J., Lin, W.: The trellis complexity of convolutional codes. IEEE Transactions on Information Theory 42(6), 1855–1864 (1996)

[27] Heideman, M.T., Burrus, C.S.: Multiplicative Complexity, Convolution, and the DFT. Springer, ??? (1988)

[28] Jose, J.M.: Optimizing neural network energy efficiency through low-rank factorisation and pde-driven dense layers. International Journal of Research Publication and Reviews 2(2), 5483–5487 (2025). ISSN: 2022

[29] Bhalgaonkar, S., Munot, M., *et al.*: Model

compression of deep neural network architectures for visual pattern recognition: Current status and future directions. Computers and Electrical Engineering 116, 109180 (2024)

[30]Dhilleswararao, P., Boppu, S., Manikandan, M.S., Cenkeramaddi, L.R.: Effi- cient hardware architectures for accelerating deep neural networks: Survey. IEEE access 10, 131788–131828 (2022)

[31]Jose, J.M.: Edge intelligence: Architecture, scope and applications. International Journal of Research Publication and Reviews 2(2), 5 (2022)

[32]Chen, Y., Zheng, B., Zhang, Z., Wang, Q., Shen, C., Zhang, Q.: Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions. ACM Computing Surveys (CSUR) 53(4), 1–37 (2020)

[33]Liu, D., Kong, H., Luo, X., Liu, W., Subramaniam, R.: Bringing ai to edge: From deep learning's perspective. Neurocomputing 485, 297–320 (2022)

[34]Lee, J., Mukhanov, L., Molahosseini, A.S., Minhas, U., Hua, Y., Rincon, J., Dichev, K., Hong, C.-H., Vandierendonck, H.: Resource-efficient convolutional networks: A survey on model-, arithmetic-, and implementation-level techniques. ACM Computing Surveys 55(13s), 1–36 (2023)

[35]Jia, W., Sun, M., Lian, J., Hou, S.: Feature dimensionality reduction: a review. Complex & Intelligent Systems 8(3), 2663–2693 (2022)

[36]He, Y., Xiao, L.: Structured pruning for deep convolutional neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)

[37]Rokh, B., Azarpeyvand, A., Khanteymoori, A.: A comprehensive survey on model quantization for deep neural networks in image classification. ACM Transactions on Intelligent Systems and Technology 14(6), 1–50 (2023)

[38]Chitty-Venkata, K.T., Somani, A.K.: Neural architecture search survey: A hardware perspective. ACM Computing Surveys 55(4), 1–36 (2022)

[39]Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., *et al.*: Searching for mobilenetv3. In: Pro- ceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1314–1324 (2019)

[40]Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neu- ral networks. In: International Conference on Machine Learning, pp. 6105–6114 (2019). PMLR

[41]Jindal, A., Jose, J.M., Benedict, S., Gerndt, M.: Lora-powered energy-efficient object detection mechanism in edge computing nodes. In: 2022 Sixth Interna- tional Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I- SMAC), pp. 237–244. IEEE, ??? (2022). https://doi.org/10.1109/I-SMAC55078. 2022.9984414 . https://doi.org/10.1109/I-SMAC55078.2022.9984414

[42]Lange, E.O., Jose, J.M., Benedict, S., Gerndt, M.: Automated energy modeling framework for microcontroller-based edge computing nodes. In: International Conference on Advanced Network Technologies and Intelligent Computing, pp. 422–437. Springer, ??? (2022). https://doi.org/10.1007/978-3-031-27082-5 32 . https://doi.org/10.1007/978-3-031-27082-5$_3$2

[43]Hong, H.S., Kim, H.: Implementation of tiled point-wise convolution in mobilenet for parallel processing. In: 2024 International Conference on Electron- ics, Information, and Communication (ICEIC), pp. 1–2 (2024). IEEE

[44]Chen, T., Tan, Y.-a., Zhang, Z., Luo, N., Li, B., Li, Y.: Dataflow optimization with layer-wise design variables estimation method for enflame cnn accelerators. Journal of Parallel and Distributed Computing 189, 104869 (2024)

[45]Kumar, S., Kulkarni, A., Akhtar, M.S., Chakraborty, T.: When did you become so smart, oh wise one?! sarcasm explanation in multi-modal multi-party dialogues. arXiv preprint arXiv:2203.06419 (2022)

[46] Bedi, M., Kumar, S., Akhtar, M.S., Chakraborty, T.: Multi-modal sarcasm detec- tion and humor classification in code-mixed conversations. IEEE Transactions on Affective Computing

14(2), 1363–1375 (2021)

[47] Hao, H., Bartusiak, E.R., Güera, D., Mas Montserrat, D., Baireddy, S., Xiang, Z., Yarlagadda, S.K., Shao, R., Horváth, J., Yang, J., *et al.*: Deepfake detection using multiple data modalities. In: Handbook of Digital Face Manipulation and Detection: From DeepFakes to Morphing Attacks, pp. 235–254. Springer, ??? (2022)

[48] Koenecke, A., Nam, A., Lake, E., Nudell, J., Quartey, M., Mengesha, Z., Toups, C., Rickford, J.R., Jurafsky, D., Goel, S.: Racial disparities in automated speech recognition. Proceedings of the National Academy of Sciences 117(14), 7684– 7689 (2020)

[49] Hiremath, B.N., Patil, M.M.: Sarcasm detection using cognitive features of visual data by learning model. Expert Systems with Applications 184, 115476 (2021)

[50] Karthik, E., Sethukarasi, T.: Sarcastic user behavior classification and prediction from social media data using firebug swarm optimization-based long short-term memory. The Journal of Supercomputing, 1–25 (2022)

[51] Li, Y., Li, Y., Zhang, S., Liu, G., Chen, Y., Shang, R., Jiao, L.: An attention- based, context-aware multimodal fusion method for sarcasm detection using inter-modality inconsistency. Knowledge-Based Systems 287, 111457 (2024)

[52] Gedela, R.T., Baruah, U., Soni, B.: Deep contextualised text representation and learning for sarcasm detection. Arabian Journal for Science and Engineering 49(3), 3719–3734 (2024)

[53] Kumar, S.K., Jose, J.M.: A survey on synthesizing images with generative adver- sial networks. International Journal of Research Publication and Reviews 2(2), 5 (2021)

[54] Galal, M.A., Yousef, A.H., Zayed, H.H., Medhat, W.: Arabic sarcasm detec- tion: An enhanced fine-tuned language model approach. Ain Shams Engineering Journal, 102736 (2024)

[55] Wang, Z., Hu, S.-j., Liu, W.-d.: Product feature sentiment analysis based on gru- cap considering chinese sarcasm recognition. Expert Systems with Applications 241, 122512 (2024)

[56] Tiwari, P., Zhang, L., Qu, Z., Muhammad, G.: Quantum fuzzy neural network for multimodal sentiment and sarcasm detection. Information Fusion 103, 102085 (2024)

[57] Yao, F., Sun, X., Yu, H., Zhang, W., Liang, W., Fu, K.: Mimicking the brain's cognition of sarcasm from multidisciplines for twitter sarcasm detection. IEEE Transactions on Neural Networks and Learning Systems 34(1), 228–242 (2023) https://doi.org/10.1109/TNNLS.2021.3093416

[58] Yue, T., Mao, R., Wang, H., Hu, Z., Cambria, E.: Knowlenet: Knowledge fusion network for multimodal sarcasm detection. Information Fusion 100, 101921 (2023)

[59] Subramanian, J., Sridharan, V., Shu, K., Liu, H.: Exploiting emojis for sarcasm detection. In: Social, Cultural, and Behavioral Modeling: 12th International Con- ference, SBP-BRiMS 2019, Washington, DC, USA, July 9–12, 2019, Proceedings 12, pp. 70–80 (2019). Springer

[60] Ouyang, K., Jing, L., Song, X., Liu, M., Hu, Y., Nie, L.: Sentiment-enhanced graph-based sarcasm explanation in dialogue. arXiv preprint arXiv:2402.03658 (2024)

[61] Bhosale, S., Chaudhuri, A., Williams, A.L.R., Tiwari, D., Dutta, A., Zhu,X., Bhattacharyya, P., Kanojia, D.: Sarcasm in sight and sound: Benchmark- ing and expansion to improve multimodal sarcasm detection. arXiv preprint arXiv:2310.01430 (2023)

[62] Bandyopadhyay, D., Kumari, G., Ekbal, A., Pal, S., Chatterjee, A., BN, V.: A knowledge infusion based multitasking system for sarcasm detection in meme.In: European Conference on Information Retrieval, pp. 101–117 (2023). Springer

[63] Yao, F., Sun, X., Yu, H., Zhang, W., Liang, W., Fu, K.: Mimicking the brain's cognition of sarcasm from multidisciplines for twitter sarcasm detection. IEEE

Transactions on Neural Networks and Learning Systems 34(1), 228–242 (2021)
[64] Justo, R., Corcoran, T., Lukin, S.M., Walker, M., Torres, M.I.: Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web. Knowledge-Based Systems 69, 124–133 (2014)
[65] Liu, P., Chen, W., Ou, G., Wang, T., Yang, D., Lei, K.: Sarcasm detection in social media based on imbalanced classification. In: Web-Age Information Management: 15th International Conference, WAIM 2014, Macau, China, June 16-18, 2014. Proceedings 15, pp. 459–471 (2014). Springer
[66] Bouazizi, M., Ohtsuki, T.O.: A pattern-based approach for sarcasm detection on twitter. IEEE Access 4, 5477–5488 (2016)
[67] Mukherjee, S., Bala, P.K.: Sarcasm detection in microblogs using na¨ıve bayes and fuzzy clustering. Technology in Society 48, 19–27 (2017)
[68] Radhakrishnan, V., Joseph, C., Chandrasekaran, K.: Sentiment extraction from naturalistic video. Procedia computer science 143, 626–634 (2018)
[69] Das, D., Clark, A.J.: Sarcasm detection on facebook: A supervised learning approach. In: Proceedings of the 20th International Conference on Multimodal Interaction: Adjunct, pp. 1–5 (2018)
[70] Ren, Y., Ji, D., Ren, H.: Context-augmented convolutional neural networks for twitter sarcasm detection. Neurocomputing 308, 1–7 (2018)
[71] Ghosh, D., Fabbri, A.R., Muresan, S.: Sarcasm analysis using conversation context. Computational Linguistics 44(4), 755–792 (2018)
[72] Nguyen, T.-L., Kavuri, S., Lee, M.: A multimodal convolutional neuro-fuzzy network for emotion understanding of movie clips. Neural Networks 118, 208– 219 (2019)
[73] Majumder, N., Poria, S., Peng, H., Chhaya, N., Cambria, E., Gelbukh, A.: Sentiment and sarcasm classification with multitask learning. IEEE Intelligent Systems 34(3), 38–43 (2019)
[74] Ren, L., Xu, B., Lin, H., Liu, X., Yang, L.: Sarcasm detection with sentiment semantics enhanced multi-level memory network. Neurocomputing 401, 320–326 (2020)
[75] Sonawane, S.S., Kolhe, S.R.: Tcsd: term co-occurrence based sarcasm detection from twitter trends. Procedia Computer Science 167, 830–839 (2020)
[76] Diao, Y., Lin, H., Yang, L., Fan, X., Chu, Y., Xu, K., Wu, D.: A multi-dimension question answering network for sarcasm detection. IEEE Access 8, 135152–135161 (2020)
[77] Jain, D., Kumar, A., Garg, G.: Sarcasm detection in mash-up language using soft-attention based bi-directional lstm and feature-rich cnn. Applied Soft Computing 91, 106198 (2020)
[78] Potamias, R.A., Siolas, G., Stafylopatis, A.-G.: A transformer-based approach to irony and sarcasm detection. Neural Computing and Applications 32(23), 17309–17320 (2020)
[79] Li, J., Pan, H., Lin, Z., Fu, P., Wang, W.: Sarcasm detection with common- sense knowledge. IEEE/ACM Transactions on Audio, Speech, and Language Processing 29, 3192–3201 (2021)
[80] Wu, Y., Zhao, Y., Lu, X., Qin, B., Wu, Y., Sheng, J., Li, J.: Modeling incongruity between modalities for multimodal sarcasm detection. IEEE MultiMedia 28(2), 86–95 (2021)
[81] Shrivastava, M., Kumar, S.: A pragmatic and intelligent model for sarcasm detection in social media text. Technology in Society 64, 101489 (2021)
[82] Eke, C.I., Norman, A.A., Shuib, L.: Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and bert model. IEEE Access 9, 48501–48518 (2021)
[83] Liang, B., Lou, C., Li, X., Gui, L., Yang, M., Xu, R.: Multi-modal sarcasm detection with interactive in-modal and cross-modal graphs. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 4707–4715 (2021)
[84] Kumar, A.: Contextual semantics using hierarchical attention network for sentiment

classification in social internet-of-things. Multimedia Tools and Appli- cations 81(26), 36967–36982 (2022)
[85] Frenda, S., Cignarella, A.T., Basile, V., Bosco, C., Patti, V., Rosso, P.: The unbearable hurtfulness of sarcasm. Expert Systems with Applications 193, 116398 (2022)
[86] Du, Y., Li, T., Pathan, M.S., Teklehaimanot, H.K., Yang, Z.: An effective sar- casm detection approach based on sentimental context and individual expression habits. Cognitive Computation 14(1), 78–90 (2022)
[87] Kamal, A., Abulaish, M.: Cat-bigru: Convolution and attention with bi-directional gated recurrent unit for self-deprecating sarcasm detection. Cognitive computation 14(1), 91–109 (2022)
[88] Jain, D.K., Kumar, A., Sangwan, S.R.: Tana: The amalgam neural architecture for sarcasm detection in indian indigenous language combining lstm and svm with word-emoji embeddings. Pattern Recognition Letters 160, 11–18 (2022)
[89] Wen, Z., Gui, L., Wang, Q., Guo, M., Yu, X., Du, J., Xu, R.: Sememe knowledge and auxiliary information enhanced approach for sarcasm detection. Information Processing & Management 59(3), 102883 (2022)
[90] Zhang, Y., Wang, J., Liu, Y., Rong, L., Zheng, Q., Song, D., Tiwari, P., Qin, J.: A multitask learning model for multimodal sarcasm, sentiment and emotion recognition in conversations. Information Fusion 93, 282–301 (2023)
[91] Kumar, A., Sangwan, S.R., Singh, A.K., Wadhwa, G.: Hybrid deep learning model for sarcasm detection in indian indigenous language using word-emoji embeddings. ACM Transactions on Asian and Low-Resource Language Informa- tion Processing 22(5), 1–20 (2023)
[92] Misra, R., Arora, P.: Sarcasm detection using news headlines dataset. AI Open 4, 13–18 (2023)
[93] Wen, C., Jia, G., Yang, J.: Dip: Dual incongruity perceiving network for sarcasm detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2540–2550 (2023)
[94] Meng, J., Zhu, Y., Sun, S., Zhao, D.: Sarcasm detection based on bert and attention mechanism. Multimedia Tools and Applications, 1–20 (2023)
[95] Sukhavasi, V., Dondeti, V.: Effective automated transformer model based sar-casm detection using multilingual data. Multimedia Tools and Applications, 1–32 (2023)
[96] Gedela, R.T., Baruah, U., Soni, B.: Deep contextualised text representation and learning for sarcasm detection. Arabian Journal for Science and Engineering, 1–16 (2023)
[97] Zhang, Y., Jia, A., Wang, B., Zhang, P., Zhao, D., Li, P., Hou, Y., Jin, X., Song, D., Qin, J.: M3gat: A multi-modal, multi-task interactive graph attention network for conversational sentiment analysis and emotion recognition. ACM Transactions on Information Systems 42(1), 1–32 (2023)
[98] Jose, J.M., Benedict, S.: Deepasd framework: A deep learning-assisted automatic sarcasm detection in facial emotions. In: 2023 8th International Conference on Communication and Electronics Systems (ICCES), pp. 998–1004 (2023). IEEE
[99] Pandey, R., Singh, J.P.: Bert-lstm model for sarcasm detection in code-mixed social media post. Journal of Intelligent Information Systems 60(1), 235–254 (2023)
[100] Zhang, Y., Yu, Y., Wang, M., Huang, M., Hossain, M.S.: Self-adaptive repre- sentation learning model for multi-modal sentiment and sarcasm joint analysis. ACM Transactions on Multimedia Computing, Communications and Applica- tions 20(5), 1–17 (2024)
[101] Tan, H., Tong, G., Huang, L., Xiao, L., Xiao, N.: Multiple-mode-supporting floating-point fma unit for deep learning processors. IEEE Transactions on Very Large-Scale Integration (VLSI) Systems 31(2), 253–266 (2022)
[102] Hung, J.-M., Li, X., Wu, J., Chang, M.-F.: Challenges and trends indeveloping

nonvolatile memory-enabled computing chips for intelligent edge devices. IEEE Transactions on Electron Devices 67(4), 1444–1453 (2020)

[103] Menghani, G.: Efficient deep learning: A survey on making deep learning models smaller, faster, and better. ACM Computing Surveys 55(12), 1–37 (2023)

[104] Mittal, S.: A survey on optimized implementation of deep learning models on the nvidia jetson platform. Journal of Systems Architecture 97, 428–442 (2019)

[105] Karimzadeh, F., Raychowdhury, A.: Twofold sparsity: Joint bit-and network-level sparsity for energy-efficient deep neural network using rram based compute- in-memory. IEEE Access (2024)

[106] Benedict, S., Reddy, S.V., Bhagyalakshmi, M., Jose, J.M., Prodan, R.: Perfor- mance improvement strategies of edge-enabled social impact applications. In: 2023 International Conference on Inventive Computation Technologies (ICICT),pp. 1696–1703. IEEE,? ? ? (2023). https://doi.org/10.1109/ICICT57992.2023.10160004 https://doi.org/10.1109/ICICT57992.2023.10160004

[107] Tossoun, B., Liang, D., Cheung, S., Fang, Z., Sheng, X., Strachan, J.P., Beau- soleil, R.G.: High-speed and energy-efficient non-volatile silicon photonic mem- ory based on heterogeneously integrated memresonator. Nature Communications 15(1), 551 (2024)

[108] Yang, J., Wang, X., Jiang, Y.: Canet: Quantized neural network inference with 8-bit carry-aware accumulator. IEEE Access (2024)

[109] Dai, C., Ren, Z., Guan, L., Liu, H., Gao, M., Lu, W., Pang, Z., Peng, C., Wu,X.: A 9t-sram in-memory computing macro for boolean logic and multiply-and-accumulate operations. Microelectronics Journal 144, 106087 (2024)

[110] Chen, J., Jun, S.-W., Hong, S., He, W., Moon, J.: Eciton: Very low-power recurrent neural network accelerator for real-time inference at the edge. ACM Transactions on Reconfigurable Technology and Systems 17(1), 1–25 (2024)

[111] Lee, S.H., Park, K., Cho, S., Lee, H.-S., Choi, K., Cho, N.I.: Fast on-device learning framework for single-image super-resolution. IEEE Access (2024)

[112] Yang, D., He, N., Hu, X., Yuan, Z., Yu, J., Xu, C., Jiang, Z.: Post-training quantization for re-parameterization via coarse & fine weight splitting. Journal of Systems Architecture 147, 103065 (2024)

[113] Zheng, Y., Wu, C., Cai, P., Zhong, Z., Huang, H., Jiang, Y.: Tiny-ppg: A lightweight deep neural network for real-time detection of motion artifacts in photoplethysmogram signals on edge devices. Internet of Things 25, 101007 (2024)

[114] Zhang, M., Li, B., Liu, H., Zhao, C.: Federated learning for radar gesture recognition based on spike timing dependent plasticity. IEEE Transactions on Aerospace and Electronic Systems, 1–14 (2024) https://doi.org/10.1109/TAES.2024.3353148

[115] Khan, Z.Y., Niu, Z.: Cnn with depthwise separable convolutions and combined kernels for rating prediction. Expert Systems with Applications 170, 114528 (2021)

[116] Chu, S.-C., Dou, Z.-C., Pan, J.-S., Kong, L., Sn áš e l, V., Watada, J.: Dwsr: an architecture optimization framework for adaptive super-resolution neural networks based on meta-heuristics. Artificial Intelligence Review 57(2), 23 (2024)

[117] Wang, Z., Yao, J., Xu, M., Jiang, M., Su, J.: Transformer-based network with temporal depthwise convolutions for semg recognition. Pattern Recognition 145, 109967 (2024)

[118] Bui, D.C., Song, B., Kim, K., Kwak, J.T.: Dax-net: a dual-branch dual-task adaptive cross-weight feature fusion network for robust multi-class cancer classi- fication in pathology images. Computer Methods and Programs in Biomedicine, 108112 (2024)

[119] Li, K., Song, Y., Zhu, X., Zhang, L.: A severity estimation method for lightweight cucumber leaf disease based on dm-bisenet.

Information Processing in Agriculture (2024)
[120]Qiao, X., Yang, Y., Xue, C., He, Y., Cui, X., Jia, S., Wang, Y.: An edram
[121]based computing-in-memory macro with full-valid-storage and channel-wise-parallelism for depthwise neural network. IEEE Transactions on Circuits and Systems II: Express Briefs, 1–1 (2024) https://doi.org/10.1109/TCSII.2024.3375319
[122]Deshmukh, S.V., Zulpe, N.S.: An optimized deep learning based depthwise separable mobilenetv3 approach for automatic finger vein recognition system. Multimedia Tools and Applications, 1–29 (2024)
[123]Cho, H.: Risa: A reinforced systolic array for depthwise convolutions and embed- ded tensor reshaping. ACM Transactions on Embedded Computing Systems (TECS) 20(5s), 1–20 (2021)
[124]Cao, H., Cooper, D.G., Keutmann, M.K., Gur, R.C., Nenkova, A., Verma, R.: Crema-d: Crowd-sourced emotional multimodal actors dataset. IEEE transac- tions on affective computing 5(4), 377–390 (2014)
[125]Abbas, S., Ojo, S., Al Hejaili, A., Sampedro, G.A., Almadhor, A., Zaidi, M.M., Kryvinska, N.: Artificial intelligence framework for heart disease classification from audio signals. Scientific Reports 14(1), 3123 (2024)
[126]Qureshi, S.A., Hussain, L., Rafique, M., Sohail, H., Aman, H., Abbas, S.R., Basit, M.A., Khalid, M.I.: Eml-psp: A novel ensemble machine learning-based physical security paradigm using cross-domain ultra-fused feature extraction with hybrid data augmentation scheme. Expert Systems with Applications 243, 122863 (2024)
[127]Sujeesha, A., Mala, J., Rajan, R.: Automatic music mood classification using multi-modal attention framework. Engineering Applications of Artificial Intelli- gence 128, 107355 (2024)
[128]Schmid, F., Koutini, K., Widmer, G.: Dynamic convolutional neural networks as efficient pre-trained audio models. IEEE/ACM Transactions on Audio, Speech, and Language Processing (2024)
[129]Zhao, J., Mao, X., Chen, L.: Speech emotion recognition using deep 1d & 2d cnn lstm networks. Biomedical signal processing and control 47, 312–323 (2019)
[130]Sultana, S., Iqbal, M.Z., Selim, M.R., Rashid, M.M., Rahman, M.S.: Bangla speech emotion recognition and cross-lingual study using deep cnn and blstm networks. IEEE Access 10, 564–578 (2021)
[131]R., M., Swamy, S.: Voice based sarcasm detection in kannada language. Interna- tional Journal of Intelligent Systems and Applications in Engineering 12(14S), (2024). Research Scholar, Department of Computer Science and Engineering, Sir
M. Visvesvaraya Institute of Technology, Visvesvaraya Technological University,
Belagavi, India; Professor, Department of Computer Science and Engineer- ing, Sir M. Visvesvaraya Institute of Technology, Visvesvaraya Technological University, Belagavi, India
[132]Gao, X., Nayak, S., Coler, M.: Deep CNN-based Inductive Transfer Learning for Sarcasm Detection in Speech. In: Proc. Interspeech 2022, pp. 2323–2327 (2022). https://doi.org/10.21437/Interspeech.2022-11323
[133]Jain, A., Patil, P., Masud, G., Krishnan, S., Jagan, V.B.: Detection of sarcasm through tone analysis on video and audio files: A comparative study on ai models performance. SSRG International Journal of Computer Science and Engineering 8(12), 1–5 (2021)