Emotion Detection Using Neural Networks on Embedded Systems

Enrico Zanetti
Department of Industrial Engineering
University of Trento
ID: 238640
Trento, Italy
enrico.zanetti-1@studenti.unitn.it
enricozanetti1998@gmail.com

***Abstract*—Emotion detection through facial expressions has gained significant attention due to its applications in human-computer interaction, security, and healthcare. This report presents a project that optimizes a neural network for real-time emotion detection using the OpenMV Cam H7 Plus, a resource-constrained edge device, and the Edge Impulse platform. The project involved setting up the hardware and software environments, collecting and preprocessing the FER2013 dataset of facial images, applying transfer learning to adapt a pre-trained model, and optimizing it for the OpenMV Cam H7 Plus. The system was tested and achieved an accuracy of 51.0% in classifying emotions. Face detection was integrated to improve accuracy by focusing on relevant image regions. Real-time data transmission and visualization were implemented using UART communication protocol, MQTT and Grafana, demonstrating potential for continuous monitoring of emotional states. Data were then fetched from the broker by using Node Red and transmitted to a InfluxDB database for persistent storage. The project showcases the feasibility of deploying emotion detection on edge devices and highlights areas for future improvement, such as enhancing model accuracy, real-time feedback, security, and integration with additional sensors.**

***Index Terms*—Emotion Detection, Edge AI, OpenMV Cam H7 Plus, Neural Network Optimization, Real-Time Monitoring, Transfer Learning, Human-Computer Interaction.**

I. Introduction

Emotion detection through facial expressions has gained significant attention in recent years due to its applications in various fields such as human-computer interaction, security, and healthcare [1], [2]. The advent of edge AI and advancements in embedded systems have made it feasible to perform complex tasks such as emotion recognition on low-power, resource-constrained devices [3], [4].

The OpenMV Cam H7 Plus, equipped with a powerful microcontroller and camera module, presents a suitable platform for implementing real-time emotion detection systems [5]. Combined with the capabilities of Edge Impulse, a platform for building and deploying machine learning models on edge devices, this project explores the optimization of a neural network to accurately detect emotions from facial images captured by the camera [6].

Previous studies have demonstrated the effectiveness of transfer learning in enhancing the performance of emotion detection models [7], [8]. By utilizing pre-trained models and fine-tuning them with a custom dataset, the neural network can achieve high accuracy while maintaining efficiency. Additionally, integrating face detection algorithms ensures that the emotion detection focuses on the relevant regions of the images, further improving the model's performance [9].

This project encompasses the following key phases:

- Preparation of the work environment, including setting up the OpenMV Cam H7 Plus and configuring the Edge Impulse platform.
- Collection and preprocessing of a comprehensive dataset of facial images, consisting a publicly available dataset downloaded from Kaggle [10].
- Application of transfer learning to adapt a pre-trained model for emotion detection using the collected dataset.
- Optimization of the model to run efficiently on the OpenMV Cam H7 Plus.
- Running the fine-tuned model on the OpenMV Cam to perform emotion detection.
- Integration of face detection to enhance emotion detection accuracy.
- Configuration of data transmission via UART communication protocol and MQTT broker.
- Storing the results in an InfluxDB database and visualizing them using Grafana, that shows an emoji that corresponds to the detected emotion.

Figure 1 illustrates the project's pipeline, detailing the various steps and technologies utilized.

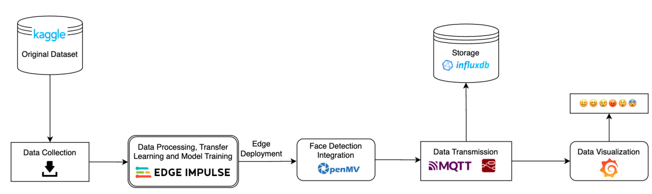


Fig. 1. The project pipeline for emotion detection

When applied in practice by following all the steps, the importance of edge AI is proven since it will help develop a practical real-time emotion detection system.

II. Importance of Emotion Detection

Emotion detection is an emerging field that plays a crucial role in various applications, ranging from human-computer interaction to healthcare and security. Understanding human emotions can significantly enhance the interaction between humans and machines, leading to more intuitive and responsive systems.

A. Applications of Emotion Detection

The technology proves essential in the following areas:

- **Human-Computer Interaction (HCI):** Enhancing user experience by allowing systems to respond to the emotional state of users. This can lead to more personalized and engaging interactions [11].
- **Healthcare:** Assisting in the monitoring and diagnosis of mental health conditions by analyzing emotional states over time [12].
- **Security:** Enhancing security systems by detecting stress or anxiety that could indicate suspicious behavior [13].
- **Marketing:** Analyzing consumer reactions to products and advertisements to understand consumer preferences and improve marketing strategies [14].
- **Education:** Adapting educational content and methods to the emotional state of students to improve learning outcomes [15].

B. Related Works

Several studies have explored the use of machine learning and deep learning techniques for emotion detection. Picard's foundational work on affective computing laid the groundwork for understanding and developing systems that can recognize and respond to human emotions [16].

Calvo and D'Mello reviewed affect detection in human-computer interaction, emphasizing the potential of combining physiological and behavioral signals for improved accuracy [12]. More recently, advancements in deep learning have led to significant improvements in emotion detection. For instance, Lane et al. demonstrated the potential of deploying deep learning models on mobile devices for real-time emotion recognition [17].

In the context of transfer learning, Tajbakhsh et al. highlighted the benefits of fine-tuning pre-trained convolutional neural networks for medical image analysis, which can be applied similarly to emotion detection tasks [7]. Furthermore, Viola and Jones' work on object detection using a boosted cascade of simple features has been instrumental in developing real-time face detection algorithms, which are essential for accurate emotion detection [9].

C. Contribution of This Project

This project aims to contribute to the field of emotion detection by optimizing a neural network to run efficiently on the OpenMV Cam H7 Plus, a resource-constrained edge device. By leveraging the capabilities of Edge Impulse for model training and deployment, this project demonstrates the feasibility of performing emotion recognition tasks on low-power devices. This approach can significantly enhance the accessibility and scalability of emotion detection systems, making them viable for real-time applications in various domains. Therefore, the affordability of the hardware combined with the intuitive user experience has the potential to create a product accessible to everyone.

In summary, this project not only advances the technical aspects of emotion detection on edge devices but also opens up new possibilities for deploying such systems in real-world applications, enhancing the interaction between humans and machines in meaningful ways.

III. Methodology

This section outlines the steps and processes followed for this project. The methodology includes the preparation of the work environment, dataset collection and preprocessing, impulse design, application of transfer learning, model selection and retraining, deployment, and data transmission, storage, and visualization.

A. Work Environment Preparation

The first step involved setting up the OpenMV Cam H7 Plus and configuring the necessary development environments.

- Acquiring the OpenMV Cam H7 Plus and installing the OpenMV IDE.
- Creating a virtual environment and installing the necessary requirements to correctly run the Python scripts.
- Setting up an Edge Impulse account and configuring the development environments.
- Creating a new InfluxDB database, ready to receive data in a specified bucket.
- Connecting the database to the Grafana dashboard for the final visualization.

B. Dataset Collection

The FER2013 (Facial Expression Recognition 2013 Dataset) [18] was selected for this project and downloaded from Kaggle [10]. This dataset is well-regarded for its size, free copyright status, and high-quality images. FER2013 contains approximately 35,000 facial images of different expressions with sizes restricted to 48×48. The main labels can be divided into 7 types: Happy, Sad, Fearful, Neutral, Angry, Surprise, and Disgust. The Disgust expression has the fewest images (600), while other labels have nearly 5,000 samples each. Due to the small number of images, this emotion was discarded for this project, while keeping all the others.

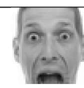
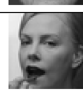
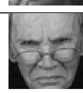
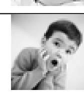
The dataset is composed entirely of grayscale images, which, instead of RGB images, offers several advantages, such as reducing computational complexity and memory usage, leading to faster training times and lower resource consumption. Grayscale images emphasize structural features over color information, which can be beneficial for tasks like facial expression recognition. Data augmentation and preprocessing are also more efficient with grayscale images. Moreover, a dataset with all images in grayscale standardizes it, reducing variability due to differing lighting conditions and color profiles, ultimately improving model performance.

After integrating the dataset, challenges emerged with Edge Impulse due to the dataset's size, which exceeded the platform's capabilities. As a result, the dataset was reduced from 35,685 images to 30,309 images. This reduction was executed carefully to maintain class balance and prevent any imbalance between the different classes.

After uploading the dataset into Edge Impulse, it was divided into training (79%) and test (21%) sets, both encompassing the six classes of emotions. Figures 2 and 3 show the number of images per class for training and test sets, respectively.

Table I provides visual representations of examples of various facial expressions associated with different emotions.

TABLE I
EXAMPLES OF DIFFERENT EMOTION CLASSES

Class	Example 1	Example 2	Example 3	Example 4
Happy				
Sad				
Fearful				
Neutral				
Angry				
Surprised				

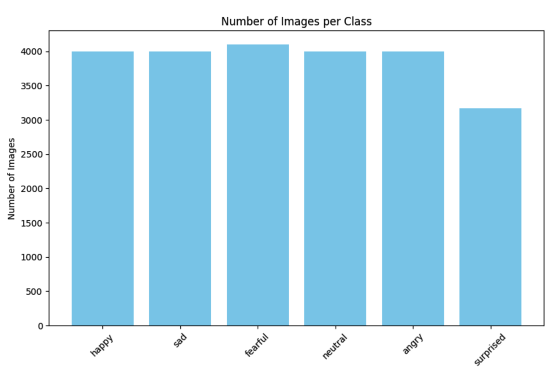


Fig. 2. Number of images per class in the Training set

C. Impulse Design

The collected images were preprocessed to ensure they were suitable for training the neural network. The preprocessing steps included the following:

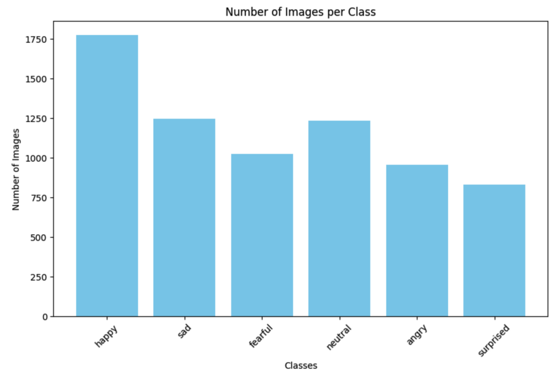


Fig. 3. Number of images per class in the Test set

- **Resizing and Normalizing:** The images of 42x42 pixels were normalized using Edge Impulse tools. These dimensions were specified during the creation of the impulse, which takes raw data, applies signal processing to extract features, and then uses a learning block to classify new data.
- **Adding Processing and Learning Blocks:** During this phase, a processing block and a learning block were added. The processing block was tailored to handle image-specific preprocessing, while the learning block was designed to perform transfer learning. In this setup, images served as input features and the respective classes as output features.
- **Configuring Image Parameters:** Subsequently, the image parameters were configured. After grayscale color depth was specified, the process of generating features began.

D. Transfer Learning

Once the generation of features was completed, transfer learning was applied to adapt a pre-trained model for emotion detection to the collected dataset.

Various neural network settings were tested with different parameter values, as shown in Table II. Ultimately, the model with the best accuracy (40.7%) was selected. The OpenMV Cam H7 Plus was set as the target device for the transfer learning phase. The model selected for this task was MobileNetV2 with a final layer of 16 neurons and a dropout rate of 0.1. This model was chosen due to its efficient architecture, which balances accuracy and computational requirements, making it suitable for deployment on the resource-constrained OpenMV Cam H7 Plus.

Through experimentation, it was observed that the optimal learning rate for this task and dataset was 0.0005. Increasing the number of epochs generally improved accuracy, and 25 training cycles were identified as the best value.

Data augmentation was applied in all experiments to enhance the model's ability to generalize, and the batch size was set to 32.

TABLE II
TRANSFER LEARNING PARAMETERS AND PERFORMANCE

Experiment	LR	Training Cycles	Accuracy (%)	Loss
1	0.0005	27	39.5	1.51
2	**0.0005**	**25**	**40.7**	**1.49**
3	0.0005	20	39.6	1.52
4	0.0005	10	38.5	1.54

E. Model Selection

The EON Tuner was employed to find and select the best embedded machine learning model for this application within the constraints of the OpenMV Cam H7 Plus. The EON Tuner analyzed the input data, potential signal processing blocks, and neural network architectures. It provided an overview of possible model architectures that fit the device's latency and memory requirements, enabling an informed selection of the optimal model configuration [19].

The process selected the grayscale-conv2d-ba8 model as the most suitable for application within the camera, and it was selected as the primary model. The confusion matrix illustrated in Fig. 4 shows the accuracy of the predictions made by the model at this stage with respect to the different classes.

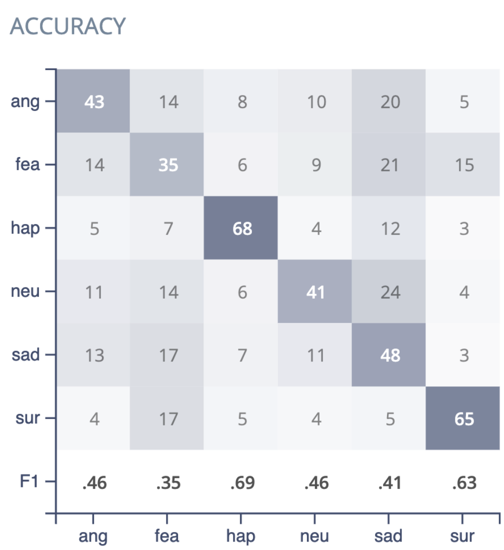


Fig. 4. Confusion matrix showing the predictions of the grayscale-conv2d-ba8 model.

F. Model Retraining

To enhance the model's performance, a retraining process was conducted. This involved using the previously selected grayscale-conv2d-ba8 model as the foundation. The retraining was carried out with the aim of improving the model's accuracy and reducing its loss.

The retrained model achieved an accuracy of 51.0% and a loss of 1.25. These results, while modest, indicate that the model is capable of performing emotion detection to a certain degree. The detailed evaluation metrics for the validation set are presented in Table III.

TABLE III
MODEL METRICS

Metric	Value
Area under ROC Curve	0.83
Weighted average Precision	0.52
Weighted average Recall	0.51
Weighted average F1 score	0.51

The AUC of this model equals 0.83, indicating good performance in differentiating classes of emotions. Precision and recall averaged by weights equal 0.52 and 0.51, while the F1 weighted average score equals 0.51. These metrics suggest that the model performs reasonably well at correctly identifying class emotion; however, there is much room for enhancement.

Given the nature of the task—to detect slight variations in facial expressions in determining emotions—these results show reasonable confidence in the model. A retrained confusion matrix is shown in Fig. 5 , showing the distribution of correct and incorrect classification of different emotion classes. This will be useful to pinpoint areas where the model needs improvement, like sadness and fear, which are similar emotions.

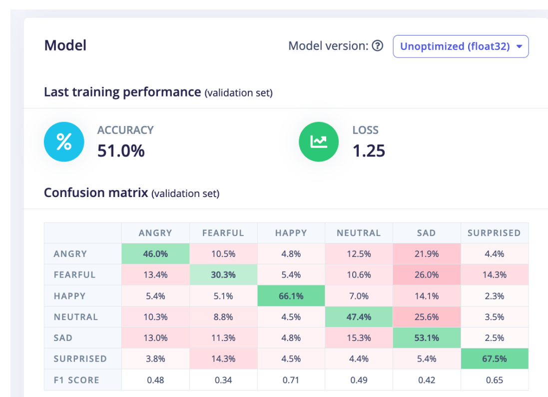


Fig. 5. Confusion matrix showing the predictions of the retrained grayscale-conv2d-ba8 model.

G. Deployment

The trained model was deployed onto the OpenMV Cam H7 Plus using the OpenMV library, which supports the deployment of machine learning models on OpenMV devices. The model was first downloaded from the Edge Impulse platform and then uploaded to the OpenMV Cam's memory. With the deployment complete, the camera is now capable of performing real-time emotion detection. A test was conducted to evaluate the capabilities of the model. To perform this test, the author's face, which was not included in the dataset used for transfer learning, was utilized. An example from this test is depicted in Fig. 6, showcasing the model's performance and its ability to accurately detect and classify emotions on previously unseen faces.

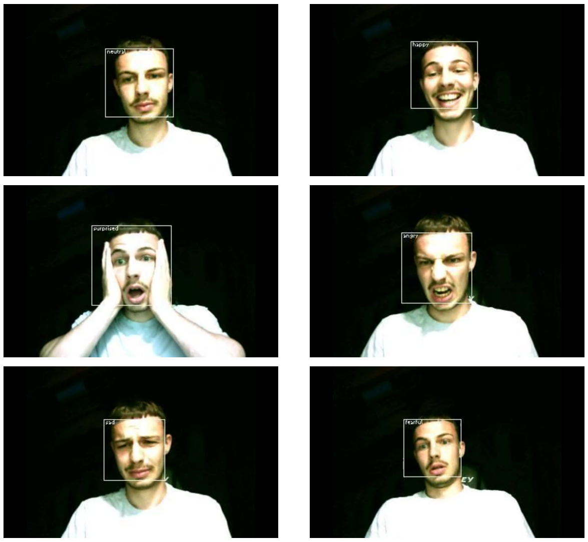


Fig. 6. Examples of classification results during the first test

H. Continuous Data Collection and Transmission Workflow

Following the deployment and testing of the model onto the OpenMV Cam H7 Plus, a continuous data collection and transmission workflow was established. This section details the scripts running at the same time to perform these operations.

1) Script on the OpenMV Camera: The first script, *main.py*, was designed to run on the OpenMV Cam H7 Plus. Its main functionalities included:

- Initialize the camera and set it to capture images in QVGA format.
- Load the pre-trained TensorFlow Lite model for emotion detection.
- Capture images and detect faces.
- Classify the emotions of the detected faces.
- Send the detected emotions via UART communication protocol.

2) Script on the PC: The second script, *openmv_emotion_mqtt_publisher.py*, was designed to run on a PC. Its main functionalities included:

- List available serial ports and select the correct one for the OpenMV camera.
- Initialize an MQTT client and connect to the specified MQTT broker.
- Continuously read emotion data from the serial port.
- Publish the detected emotions to the MQTT broker under the topic ”ezan/emotion_detection”.

I. Data Storage and Visualization

Node-RED was utilized to fetch data from the MQTT broker. This setup involved configuring Node-RED to subscribe to the relevant MQTT topic, process the incoming data, and then store it in the InfluxDB database. During data preprocessing, Node-RED associated an emoji with each detected emotion and added a timestamp. The associated emojis are shown in Table IV. The processed data was then transmitted to InfluxDB and and a Grafana dashboard.

The dashboard, Fig. 7, visualizes the real-time emotion detection data collected from the OpenMV camera. The upper section, labeled ”Emotion Detection Table,” displays a timestamped list of detected emotions, represented by corresponding emojis. The lower section, ”Emotion Detection State Timeline,” provides a timeline view of the detected emotions, allowing for an intuitive understanding of emotion transitions over time.

TABLE IV
EMOTIONS AND ASSOCIATED EMOJIS

Emotion	Emoji
Neutral	😐
Happy	😊
Surprised	😲
Angry	😡
Sad	😢
Fearful	😨

Fig. 7. Grafana dashboard showing real-time emotion detection data. The upper section displays a table of detected emotions with timestamps, and the lower section shows a timeline of emotion states, represented by emojis.

IV. RESULTS AND ANALYSIS

The results of this project demonstrate the feasibility of deploying an emotion detection neural network on a resource-constrained device such as the OpenMV Cam H7 Plus. Despite the limited computational power of the hardware, the system achieved a reasonable level of accuracy in classifying emotions. The chosen model, MobileNetV2, provided a balance between accuracy and efficiency, making it suitable for real-time applications on an embedded device.

The integration of face detection significantly enhanced the accuracy of emotion detection by ensuring that only relevant regions of the image were analyzed. This approach reduced noise and improved the model’s focus on the facial features that are most indicative of emotional states.

However, the accuracy of the emotion detection model, while reasonable, leaves room for improvement. The model achieved an accuracy of around 51.0% with a loss of 1.25, indicating that there are still challenges in distinguishing between similar emotions such as sadness and fear. The confusion matrix highlighted these areas of difficulty, suggesting that the model's performance could be enhanced with further training and refinement.

The real-time data transmission and visualization components of the system, using MQTT, Node-Red, InfluxDB, and Grafana, successfully demonstrated the potential for continuous monitoring and analysis of emotional states. The dashboard provided an intuitive interface for viewing real-time emotion detection data, which could be valuable in applications such as mental health monitoring or human-computer interaction.

V. Challenges and Limitations

Throughout the implementation process, several challenges were encountered, particularly in the data transmission phase.

During the learning phase, training was limited to 25 epochs due to the Edge Impulse account restriction, which caps transfer learning sessions at a maximum of 20 minutes. This limitation was a significant constraint to further attempts to improve the model's accuracy, as longer training sessions would likely enhance performance.

Another challenge was the dependency on the OpenMV IDE for running the camera script by using UART communication protocol, which caused serial port conflicts when attempting to read and transmit data simultaneously using two different Python scripts. This issue was resolved by making the OpenMV camera script run independently from the IDE. By saving the script directly to the camera's internal storage, it can run automatically upon power-up, freeing up the serial port for data transmission, thus avoiding the use of the OpenMV IDE. However, this introduced a significant limitation: the user can't see their face in real-time during emotion detection when this data transmission phase is used.

This issue could be addressed by using additional devices or implementing WiFi transmission using the UDP protocol. This involves configuring the OpenMV Cam H7 Plus with a WiFi shield and setting up a UDP server on a connected device to receive data packets. This method allows for efficient, low-latency data transmission, enabling real-time display of emotion detection results.

VI. Future Improvements

There are several areas where this project could be enhanced to improve the performance and applicability of the emotion detection system.

A. Model Accuracy and Training

To improve the accuracy of the emotion detection model, additional training with a larger and more diverse dataset could be beneficial. Incorporating more images with varied lighting conditions, backgrounds, and facial expressions could help the model generalize better. Additionally, experimenting with different neural network architectures and hyperparameters might yield a model with better performance.

B. Real-time Feedback and User Interface

Currently, the system does not provide real-time feedback to the user due to the limitations of running the script independently on the OpenMV Cam H7 Plus. Developing a user interface that allows for real-time display of emotion detection results could enhance the user experience. As mentioned, this could be achieved by utilizing the WiFi capabilities of the OpenMV Cam to stream video and emotion data to a connected device.

C. Enhanced Data Transmission and Security

While MQTT and InfluxDB provided an effective means of data transmission and storage, implementing enhanced security measures would be important for real-world applications. Encrypting the data transmission and ensuring secure access to the InfluxDB database would protect sensitive emotional data from potential breaches.

D. Integration with Other Sensors and Contextual Data

Integrating the emotion detection system with other sensors, such as heart rate monitors or voice analysis tools, could provide a more comprehensive understanding of emotional states. Combining facial expression data with physiological and contextual information could enhance the accuracy and reliability of emotion detection.

E. Deployment on Alternative Platforms

Exploring the deployment of the emotion detection model on other embedded platforms with greater computational power could also be a future improvement. Devices such as the NVIDIA Jetson Nano [20] or Google Coral [21] could offer more robust performance, allowing for the use of more complex models and real-time video processing capabilities.

By addressing these areas for improvement, the emotion detection system could become more accurate, reliable, and versatile, opening up new possibilities for its application in various domains.

VII. Conclusion: a Future Vision

A compelling vision for the future involves devices that are fully managed by advanced AI systems capable of adapting and adjusting their functions and modes of communication based on the user's detected emotions. A system like the one presented in this project could be implemented in this framework. Such capabilities could enhance user experience by making interactions more intuitive and responsive, creating personalized environments that cater to individual emotional states. However, the implementation of these systems must be approached with caution, ensuring that ethical considerations and respect for the natural environment are considered as priority constraints. It is essential to protect user privacy and maintain the freedom of individuals, preventing misuse of sensitive emotional data. Balancing technological advancements with ethical standards will be crucial in developing responsible and user-centric AI-driven systems.

REFERENCES

[1] P. Pranathi, C. Lakshmi, and M. Suneetha, "A review on various facial expression recognition techniques," in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 1246–1254, 2021.

[2] A. Gupta and S. Batra, "Real-time emotion recognition using deep facial expression analysis on mobile devices," *Research Journal of Computer Systems and Engineering*, vol. 4, no. 2, pp. 89–102, 2023.

[3] Y. Wei, Z. Zhong, and J. Gu, "Human emotion based real-time memory and computation management on resource-limited edge devices," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pp. 487–492, 2022.

[4] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[5] OpenMV, "Openmv cam h7 plus." https://openmv.io/products/openmv-cam-h7-plus. Accessed: 2024-06-21.

[6] Edge Impulse, "Edge impulse - machine learning platform." https://edgeimpulse.com/. Accessed: 2024-06-21.

[7] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1, pp. I–I, Ieee, 2001.

[10] "Emotion detection fer dataset." https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer/data, 2013.

[11] J. Lazar, J. H. Feng, and H. Hochheiser, *Research methods in human-computer interaction*. Morgan Kaufmann, 2017.

[12] R. A. Calvo and S. D'Mello, "Affect detection: An interdisciplinary review of models, methods, and their applications," *IEEE Transactions on affective computing*, vol. 1, no. 1, pp. 18–37, 2010.

[13] F. Agrafioti, D. Hatzinakos, and A. K. Anderson, "Ecg pattern analysis for emotion detection," *IEEE Transactions on affective computing*, vol. 3, no. 1, pp. 102–115, 2011.

[14] D. McDuff, R. Kaliouby, T. Senechal, M. Amr, J. Cohn, and R. Picard, "Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 881–888, 2013.

[15] R. S. Baker, S. K. D'Mello, M. M. T. Rodrigo, and A. C. Graesser, "Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments," *International Journal of Human-Computer Studies*, vol. 68, no. 4, pp. 223–241, 2010.

[16] R. W. Picard, "Affective computing for hci.," in *HCI (1)*, pp. 829–833, Citeseer, 1999.

[17] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, L. Jiao, L. Qendro, and F. Kawsar, "Deepx: A software accelerator for low-power deep learning inference on mobile devices," in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 1–12, IEEE, 2016.

[18] I. Goodfellow *et al.*, "Fer2013 dataset." https://paperswithcode.com/dataset/fer2013, 2013. Accessed: 2024-07-01.

[19] Edge Impulse, "Introducing the eon tuner: Edge impulse's new automl tool for embedded machine learning." https://www.edgeimpulse.com/blog/introducing-the-eon-tuner-edge-impulses-new-automl-tool-for-embedded-machine-learning/. Accessed on: June 28, 2024.

[20] NVIDIA, "Jetson nano developer kit product development." https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/. Accessed: 2024-07-02.

[21] Coral, "Coral ai." https://coral.ai/. Accessed: 2024-07-02.