ISSN 2347 - 3983
Volume 11. No.12, December 2023
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter0711122023.pdf
https://doi.org/10.30534/ijeter/2023/07111122023

Human Face Classification using TensorFlow and Deployment onto ASIC

V Uma[1], B Sreehitha[2], Ch Nainitha[3], S Laxmi Keerthana[4], N Laxmi Aishwarya[5]
[1]G Narayanamma Institute of Technology and Science (for Women), India, uma.volety@gnits.ac.in
[2]G Narayanamma Institute of Technology and Science (for Women), India, sreehithabotta@gmail.com
[3]G Narayanamma Institute of Technology and Science (for Women), India, chittalurinainitha@gmail.com
[4]G Narayanamma Institute of Technology and Science (for Women), India, keerthi242003@gmail.com
[5]G Narayanamma Institute of Technology and Science (for Women), India, nethikarlaxmiaishwarya@gmail.com

Received Date: October 20, 2023 Accepted Date: November 25, 2023 Published Date : December 07, 2023

ABSTRACT

This paper aims to develop a human face classification system using TensorFlow and deploying it onto ASIC for Biometrics applications. The Convolutional Neural Networks (CNN) Algorithm is used to classify human faces into predefined categories such as age, gender, and emotion. The CNN model will be trained using a large dataset of labelled images, and the training process will be optimized for ASIC deployment. The trained model will be deployed on an ASIC chip, which is optimized for power and speed.

The large dataset will be tested for accuracy and efficiency, and its performance will be evaluated in various engineering applications, such as Security, Biometrics, and Entertainment. The project will demonstrate the feasibility of using TensorFlow Lite and ASIC for developing efficient and accurate human face classification systems for Biometrics applications.

Key words: CNN Algorithm, Human faces with labelled dataset, Maix Dock, MaixPy IDE, Python.

1 INTRODUCTION

Human Face classification is the method through which a visual system recognizes the appropriate person's face. Due to its application in security systems, access control, video surveillance, commercial spaces, and even social networks like Facebook, it has become a crucial tool for human-computer interaction. Face recognition is the main method of identifying a human, and it has acquired favor again after artificial intelligence grew so quickly. It is also non-intrusive compared to other biometric techniques.

It is easy to check facial recognition in an uncontrolled environment without the subject's knowledge. TensorFlow is a useful tool in modern applications that makes it easier to identify facial traits in photos and facilitates customised user experiences, emotion analysis, and authentication. One such application is human face classification. Convolutional Neural Networks (CNNs), one type of deep learning method, have greatly increased the robustness and effectiveness of facial categorization systems.

TensorFlow, developed by Google, has emerged as a widely adopted platform for developing and training deep learning models, providing researchers and developers with a rich set of tools and resources.

Creating and refining models for a variety of applications, such as image classification, is made possible by the well-known open-source machine learning framework TensorFlow. Once you've identified the facial features in an image, you can use TensorFlow to classify the image based on a set of predefined classes. TensorFlow provides a range of pre-trained models that you can use for this purpose, or you can create your own custom models using your own data. The popular open-source machine learning framework TensorFlow makes it simple to build and improve models for a range of applications, including image classification.

Proposed system uses CNN algorithm which is a type of neural network with hidden convolution layers. The CNN would be trained to identify facial features such as eyes, nose, and mouth, and then use these features to classify the image into one of several categories, such as happy, sad, or angry. Once the CNN has been trained, you can use it to classify new images that it has not seen before. To implement human face classification using TensorFlow, Collect a large dataset of labeled images of human faces. Preprocess the images to ensure that they are all the same size. Use the trained model to identify new photos of human faces and assess its accuracy and good data generalization by comparing its performance on a different test dataset.

Application-Specific Integrated Circuit (ASIC) are specialized integrated circuits designed for specific applications, offering high efficiency and performance for dedicated tasks. By deploying face classification models onto ASIC, we can achieve accelerated and energy-efficient processing, making it ideal for applications like smart surveillance systems, facial recognition in mobile devices, and more.

This paper presents a comprehensive study on the development and deployment of a human face classification model using TensorFlow, a popular deep learning framework, onto an Application-Specific Integrated Circuit (ASIC). Our research aims to bridge the gap between software-based machine learning models and custom hardware acceleration, ultimately striving for optimized real-time performance and energy efficiency.

2 LITERATURE SURVEY

[1] Mr. Rutik Sansare, Dr. Vinayak Bharadi, Mr. Vishant Shinde, Mr. Tushar Padelkar Real Time Face Recognition System Using Convolutional Neural Network.

An average recognition accuracy of 97-99% is achieved. The implementation process consists of gathering face samples, preprocessing images, training the model, and completing the face recognition task. Based on these results, it can be concluded that VGG16 convolutional networks with transfer learning are highly accurate while consuming less computational power and can run smoothly on low-end devices.

[2] R. Joshna, Mr. Bhushan Vikas Mali, K. Mounika, G. Kusuma Priya Face Image Classification Using CNN.

The model is trained with over numerous photos of people with various ethnicities, colors, and other characteristics from the Kaggle dataset. The image is fed through multiple layers by decreasing the size of each layer. Every image in the dataset has a label as well. After the model has been trained, put it to use for testing. Identify every image's human presence first. Next, use the OpenCV framework to analyse the image and extract all of the faces of people that are in it..

[3] Haiyan Liu, Shifeng Shang, Guannan Li, Jie Cao, Qiang Qu A Face Recognition Model Based on Convolutional Neural Network.

This paper presents a sample set for a global features and local features fusion of training methods. By using model training, the method can extract as many important face features as possible, including features from the mouth, nose, eyes, and other areas. It can also extract more explicit feature vectors from these face parts. According to the experiment results, eyes are more crucial for feature extraction than the nose and mouth since they are crucial for facial identification. In conclusion, the model can extract global features more effectively if local and global features are included during training.

[4] Lin Xu, Peng Lu, Baoye Song Human face recognition based on convolutional neural network and augmented dataset.

In order to increase facial picture categorization accuracy, a CNN model is created in this research. Although some model parameters such as input data, network width, and entire connection layer differ between the model and the standard LeNet-5 model, the model's structure is still identical. Convolutional layers C1 and C2 make up the constructed CNN.

[5] Yi-Leh Wu, Cheng-Yuan Tang, Chen-Chun Huang Human Face Sentiment Classification Using Synthetic Sentiment Images with Deep Convolutional Neural Networks.

In this paper, we use a deep convolutional neural network with great accuracy to recognize the sentiment on the face. The CNN model is designed to enhance the precision of facial images. First, training and testing on real face datasets yielded average testing accuracy of 88.77%, 75.78%, 89.84%, and 80.34%. Second, training on synthetic face images alone results in substantially poorer accuracy compared to training on real or mixed faces.

3 CONVOLUTIONAL NEURAL NETWORKS

Feature extraction is a procedure that uses a convolution tool to separate and identify the distinct characteristics of an image for study. There are numerous pairs of convolutional or pooling layers in the feature extraction network. a fully connected layer that makes use of the convolutional process's output and determines the class of the image using the features that were previously extracted. This CNN feature extraction model seeks to minimize the quantity of features in a dataset. It generates new features that compile an initial set of features' existing features into a single new feature. As depicted in the CNN architectural diagram, there are numerous CNN levels.

3.1 Convolutional Layer

Convolutional, pooling, and fully-connected (FC) layers are the three different types of layers that make up the CNN. A CNN architecture will result from the stacking of these layers. Two other crucial factors, the dropout layer and the activation function, are in addition to these three layers and are defined below (Figure 1).

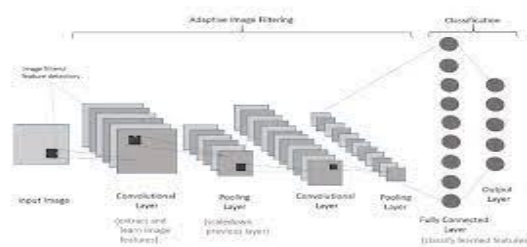

Figure 1: Architecture of CNN

1. CONVOLUTION LAYER The various features from the input photos are extracted using this layer as shown in fig 1. The input image and a filter of a specific size MxM are combined mathematically in this layer to create convolution. When the filter is moved across the input image, the dot product is calculated between the filter and the portions of the image that correspond to the filter's size (MxM). The result, known as the "Feature map," provides details about the image, including its corners and edges. Later, more layers are fed this feature map to learn additional features from the input image.

2. POOLING LAYER This layer serves as the beginning layer for the extraction of the various features from the input pictures. Convolution is created in this layer through the mathematical combination of the input image and a filter with a particular size MxM. The dot product between the filter and the areas of the input image that correspond to the filter's size (MxM) is calculated as the filter is moved over the image. The end product, dubbed the "Feature map," offers information about the image, such as its corners and edges. This feature with the three channels R, G, and B. The feature extraction from aligned faces is the most important aspect of this approach. Because CNN is employed in deep learning to work on data with grid-like structure, such as photos, feature extraction is performed using CNN map is later given additional layers to extract more features from the source image.

3. FULLY CONNECTED LAYER To connect the neurons between two layers, the Fully Connected (FC) layer, which also includes weights and biases, is utilized. These layers make up the final few layers of a CNN architecture and are often positioned before the output layer. This process flattens the input image from the preceding layers and feeds it to the FC layer. The flattened vector is then put through a few additional FC layers, where the standard operations on mathematical functions happen. The classification procedure starts to take place at this point. Because two fully connected layers will function better than one connected one, two layers are connected. These CNN layers lessen the amount of human oversight.

4 PROPOSED SYSTEM

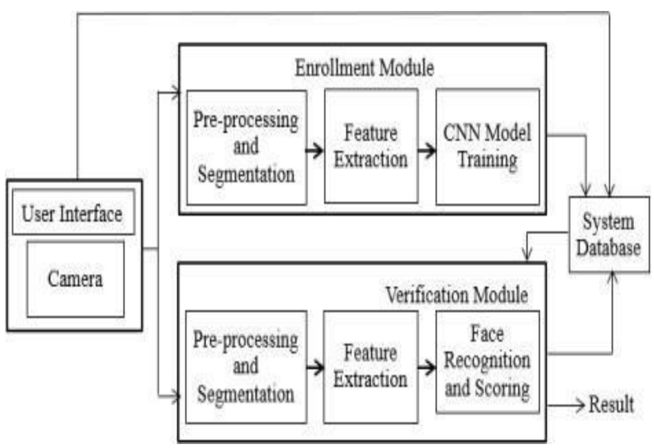

Figure 2: Block diagram of Proposed System

Figure 2 is the proposed system's workflow. Data is a centralized and crucial component of machine learning. Data from 3 people were gathered to train the CNN model. For the training set and validation dataset, 60 photos for each individual were gathered. Face identification from a given dataset is the initial stage in a face recognition system. The system will be more accurate if face classification is accurate and error-free. Face classification is a function of computer vision technology. The KPU is a hardware accelerator designed to accelerate neural network operations, particularly those used in CNNs. Faces can be detected by this classifier in colored RGB photos. An open-source library called OpenCV offers the finest environment for creating various systems that need picture pre-processing. Typically, not all of the faces that are recognized are in the correct alignment. It causes a mistake in the facial recognition procedure.

To achieve high accuracy in the face recognition process, the cropped face photos must be aligned before being sent to the neural network. We convert a specified set of points from the input image into the output coordinate system during this operation. A coordinate system for the output based on the idea of nodal points. On our faces, there are 80 different nodal points, including the contour of our cheekbones, the breadth of our nose, the length of our jawline, the space between our eyes, and more. Before feature extraction, a specific portion of the image must be removed from which we will train the CNN model. Keeping the intended aligned face and removing the undesired area of the image are included. For more accuracy, we are using a cropped face image with a 224 by 224 resolution. The train folder will hold these faces that have been cropped. The feature extraction from aligned faces is the most important aspect of this approach. Because CNN is employed in deep learning to work on data with grid-like structure, such as photos, feature extraction is performed using CNN [9]. A 128-d vector can be extracted by CNN from aligned faces. It creates a binary version of a graphical image known as a digital image. This digital image will also be utilized to train a model.

We employed the convolutional neural network's pretrained VGG16 architecture [12]. VGG16 accepts a 224 × 224-pixel input image]. A 128-d vector can be extracted by CNN from aligned faces. It creates a binary version of a graphical image known as a digital image. This digital image will also be utilized to train a model.

We employed the convolutional neural network's pretrained in MaixPy IDE. MaixPy IDE accepts a 224 × 224-pixel input image with the three channels R, G, and B. When doing face recognition, the OpenCV system helps find faces to extracts their features. We used Maixbit to deliver the labelled datasets and face classification code after deploying the datasets on SD cards. The obtained datasets were arranged into several tagged datasets and resized 50*50 into preserved datasets. Implement the code in the MaixPy IDE before deploying it to the Sipeed ASIC board. The dataset will emerge, and the specified data will be run through it. Give the input photographs; it will display an image of input effectiveness in the tagged dataset. It produces an output that is accurate and effective.

5 HARDWARE DEVELOPMENT

Training a deep learning model to categorize human faces and then deploying that model onto an ASIC (Application-Specific Integrated Circuit) device called Maix Bit are the two steps involved in human face classification using TensorFlow and deployment on ASIC (Maix Bit). A development board based on the Kendryte K210 System-on-Chip (SoC) is referred to as the "Maix Bit" hardware kit as shown in fig 3. The Maix Bit is made to offer developers interested in AI and IoT (Internet of Things) applications a portable and simple-to-use platform.

With Maix Dock, developers may take advantage of the Kendryte K210 SoC's capabilities for a variety of projects. Maix Dock is a strong and adaptable platform for prototyping and implementing edge AI applications. It is suitable for jobs that call for local AI processing and inference due to the hardware features and software support it combines. The Kendryte K210, which houses both a dedicated neural network processor (KPU) and a dual-core RISC-V processor (RV64IMAFDC), powers the Maix Dock (Figure 3). Combining these two technologies enables effective AI and machine learning inference at the edge.

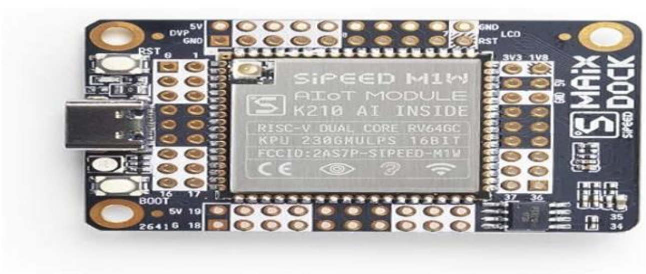

Figure 3: Maix Dock

The key components of the Maix Bit hardware kit are

Kendryte K210 SoC: The board's main processor, equipped with two RISC-V cores and AI acceleration for neural network inference. It manages program execution and data processing.

Random Access Memory (RAM): While the board is powered on, this component provides temporary storage for the processor to perform programs and store data.

USB Flash Drive: Non-volatile storage is used to store firmware, source code, and other crucial information that must survive power failures of the board.

USB port: Enables the Maix Bit to be connected to a computer for power, data transfer, and programming.

Unified Asynchronous Receiver/Transmitter (UART): Enables serial connection with external modules or other devices, such as sensors.

Inter-Integrated Circuit, or I2C: A serial communication protocol that makes it easier for the Maix Bit to communicate with peripherals and other devices, such as sensors.

Serial Peripheral Interface (SPI): An additional serial connection protocol is utilized to link the board to external components like displays and sensors.

General-purpose input/output, or GPIO, pins: Offers all-purpose input/output interfaces for connecting a range of sensors, actuators, and extraneous parts.

Interface for cameras: A connector that enables computer vision applications such as object identification and picture recognition by connecting a camera module.

Audio Connection: Enables audio input and output capabilities, which qualifies the board for applications involving audio.

Power Source: Powers the board, and it can be powered by an external power source via a USB connection.

Debugging and JTAG: The board has options for programming debugging through JTAG interfaces, enabling programmers to examine hardware.

6 SOFTWARE DEVELOPMENT

A. TensorFlow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. TensorFlow is an open-source end-to-end platform for creating Machine Learning applications. It is a symbolic math library that uses dataflow and differentiable programming to perform various tasks focused on training and inference of deep neural networks. It allows developers to create machine learning applications using various tools, libraries, and community resources.

Currently, the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations. TensorFlow is utilized in human face classification for model development, preprocessing, training, evaluation, and deployment onto Maix Bit ASIC. TensorFlow provides a comprehensive set of tools and libraries for building and deploying machine learning models, particularly deep learning models. It allows users to construct and train neural networks using high-level APIs like Keras, as well as providing low-level APIs for more advanced users who need fine-grained control over the model architecture and computations.

1. Data Collection: Gather a dataset of human face images labeled with their corresponding classes, such as different individuals or facial expressions. This dataset will be used to train and evaluate the classification model. Collected few images (154 images) belonging to 3 different classes and saved all those images into a folder. The pixels of each image is 320 X 240.

2. Preprocessing: Preprocess the face images to ensure consistency and enhance the quality of the data. This step may involve tasks like resizing the images, normalizing pixel values, and applying data augmentation techniques to increase the dataset's diversity.

3. Model Training: Use TensorFlow, a popular deep learning framework, to develop and train a face classification model. You can utilize various architectures like Convolutional Neural Networks (CNNs) for this purpose. Train the model on your dataset, optimizing it to minimize classification errors and maximize accuracy.

4. Model Evaluation: Evaluate the trained model's performance using a separate validation dataset.

5. Deployment on Maix Bit: Deploy the converted model onto the Maix Bit ASIC device. Maix Bit is specifically designed for edge AI applications and is equipped with an AI accelerator chip to efficiently run deep learning models.

6. Inference and Classification: Utilize the deployed model on the Maix Bit ASIC device to perform real-time human face classification tasks. This involves feeding the face images into the model and obtaining the predicted class labels.

7. Performance Optimization: Fine-tune the ASIC deployment to maximize performance and power efficiency, ensuring real-time processing capabilities for face classification tasks.

8. Testing and Validation: Thoroughly test the ASIC-based face classification system to verify its accuracy, speed, and power consumption against the original TensorFlow model.

9. Monitoring and Maintenance: Continuously monitor the system's performance, and if necessary, perform updates and maintenance to ensure optimal functionality.

B. MaixPy IDE

MaixPy is a Micro python-based firmware specifically developed for Sipeed's Maix series hardware. Micro python is a lightweight implementation of the Python programming language that is designed to run on microcontrollers and embedded systems. MaixPy provides an easy-to-use and efficient platform for programming and running AI applications on Sipeed's AI development boards. MaixPy is built on the AIoT K210 chip and utilizes Micro python syntax. MicroPython is an efficient implementation of Python 3, optimized to run on.

7. RESULTS

We have successfully implemented and deployed human face classification using TensorFlow onto ASIC. Human face classification using TensorFlow and deployment on ASIC (Maix Bit) involves training a deep learning model to classify human faces and then deploying that model onto an ASIC (Application-Specific Integrated Circuit) device. The proposed system achieved accuracy of 96% .

Figure 4. Input Image of Person 1

Figure 5: Output Image of Person 1

Figure 4 and figure 5 shows input and output images of a person1. The data is collected with different postures and after performing image preprocessing task are stored in trained folder and deployed on to ASIC. The Verification is done using MaixPy IDE.

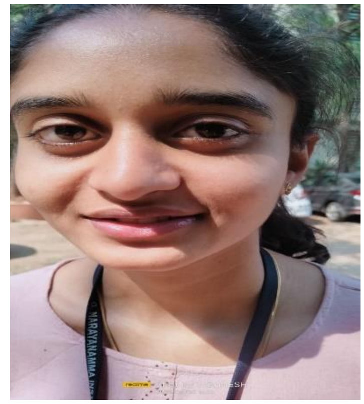

Figure 6: Input Image of Person 2

Figure 7: Output Image of Person 2

Figure 6 and figure 7 shows input and output images of a person. The data is collected with different postures and after performing image preprocessing task are stored in trained folder and deployed on to ASIC. The Verification is done using MaixPy IDE.

8. CONCLUSIONS

A human face classification method based on convolution neural network (CNN) is presented in this paper. And the network has ninty-seven layers. Human face classification using CNN achieved a maximum efficiency of 97%. The Maixpy IDE platform was employed for training and testing, leading to a robust CNN with excellent convergence.

"Human Face Classification using TensorFlow and Deployment onto ASIC" concludes with showcasing a strong and effective method for precise facial recognition. High precision is achieved by utilizing TensorFlow's deep learning capabilities for efficient feature extraction and recognition. The system is appropriate for real-time applications with low latency requirements due to the hardware acceleration provided by the ASIC deployment. Our project shows the potential of this technology in real-world applications by concentrating on improving accuracy through sophisticated architectures, data augmentation, and domain adaption, as well as tackling unconditional environment concerns. This strategy has a lot of potential for use in security, surveillance, and biometrics, among other areas, and with further research and optimization, it might help create a safer and more effective future.

REFERENCES

[1]. M. Coşkun et al., "Face recognition based on convolutional neural network", 2017 International Conference on Modern Electrical and Energy Systems (MEES), 2017.

[2]. S. F. Kak et al., "A review of person recognition based on face model", Eurasian Journal of Science & Engineering, vol. 4, no. 1, pp. 157-168, 2018.

[3]. A. K. Jain et al., Introduction to biometrics, Springer Science & Business Media, 2011.

[4]. A. Varol and B. Cebe, "Yüz Tanıma Algoritmaları Algorithms of Face Recognition", 5th International Computer & Instructional Technologies Symposium, 2011

[5]. W. Wang et al., "Face recognition based on deep learning", International Conference on Human Centered Computing, 2014.

[6]. P. Kamencay et al., "A new method for face recognition using convolutional neural network", Digital Image Processing and Computer Graphics, vol. 15, pp. 663-672, 2017.

[7]. A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, 2012.

[8]. E. Pasolli, F. Melgani, D. Tuia, F. Pacifici and E. W. J, "SVM active learning approach for image classification using spatial information", IEEE Transactions on Geoscience and Remote Sensing, 2014.

[9]. A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem", 2018 International Interdisciplinary PhD Workshop (IIPhDW), 2018.

[10]. Borkar, Neel Ramakant and Sonia Kuwelkar, "Real-Time Implementation of Face Recognition System", 2017 International Con ference on Computing Methodologies and Communication (ICCMC), 2017.

[11]. Soad Almabdy and Lamiaa Elrefaei, "Deep Convolutional Neural Network-Based Approaches for Face Recognition", Applied Sciences, 2019.

[12]. Gil Levi and Tal Hassner, "Age and gender classification using convolutional neural networks", Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015.

[13]. T. Zhang, W. Zheng, Z. Cui, Y. Zong, J. Yan and K. Yan, "A deep neural networkdriven feature learning method for multi view facial expression recog recognition", IEEE Transactions on Multimedia, vol. 18, no. 12, pp. 2528-2536, 2016.

[14]. Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin et al., "Tensorflow: Large- scale machine learning on heterogeneous distributed systems", arXiv preprint, 2016.

[15]. Jeffrey S Coffin and Darryl Ingram, Facial recognition system for security access and identification, November 1999.