Face Detection and Emotion Recognition

CSIT 697 Master's Project

Yamini Pathuri
Masters in Data Science
Department of Computer Science
Montclair State University
Montclair, NJ, USA
pathuriy1@montclair.edu

Abstract- In the rapid advancement of computer vision technologies, the fusion of face recognition and emotion detection has emerged as a compelling and multifaceted research domain. This paper delves into the integration of Convolutional Neural Networks (CNNs) with Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (Adam) optimizers to enhance the accuracy and efficiency of face recognition and emotion detection systems. Additionally, OpenCV is leveraged for precise face detection, serving as the foundational pillar for our framework.

This paper showcases a series of experiments, and the performance of our model against existing optimizers. Through rigorous testing and evaluation, I intend to provide a valuable resource for researchers, practitioners, and enthusiasts interested in advancing the capabilities of face recognition and emotion detection technology.

Keywords- Face detection; Emotion recognition, Conventional Neural Network (CNN); Stochastic Gradient Descent (SGD); Adaptive Moment Estimation (Adam); OpenCV; Optimizers;

1. Introduction

The fusion of face recognition and emotion detection has gained prominence in recent years, driven by its wide-ranging applications in diverse fields such as security, healthcare, and human-computer interaction. With the advent of deep learning techniques, Convolutional Neural Networks (CNNs) have emerged in advancing the accuracy of face recognition systems. However, to harness the full potential of CNNs, the choice of optimization algorithms is crucial.

In this context, this paper explores the use of two popular optimization algorithms: Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (Adam) in conjunction with CNNs to boost the performance of face recognition systems. These optimization algorithms not only enhance the training efficiency of CNNs but also contribute to their convergence speed and, consequently, the real-time applicability of face recognition systems.

A critical component of any face recognition system is accurate face detection, which serves as the foundation for subsequent recognition and emotion analysis. Here, we leverage the capabilities of OpenCV, a powerful computer vision library, for precise face detection. By combining OpenCV's robust face detection with our optimized CNN-based recognition framework, we aim to create a comprehensive and efficient system.

Additionally, this paper extends its focus beyond the realms of mere face recognition by incorporating emotion detection into the system. Understanding emotions from facial expressions has broad implications, from improving user experience in human-computer interaction to enhancing security measures by detecting suspicious emotional cues. This integration of emotion detection augments the versatility and practicality of our framework.

2. Related Work

In this paper [1], the author proposes a novel featuring dual-channel expression recognition based on machine learning theory and emotional philosophy. Because features extracted using CNN ignore subtle changes in the active regions of facial expressions, the proposed algorithm's first path takes the Gabor feature of the ROI region as input. This author used the dataset FER 2013 and obtained accuracy of 74%.

This paper [2], presented the research on FER, allowed us to know the latest developments. The author described different architectures such as CNN and CNN-LSTM recently proposed by different researchers, and presented some different databases containing spontaneous images collected from the real world and others formed in laboratories, in order to have and achieve an accurate detection of human emotions.

Author uses the classification algorithms used in conventional FER include SVM, Adaboost, and Random Forest, by contrast, deep learning-based FER approaches highly reduce the dependence on face physics based models. In [3] after implementing the above mentioned classifiers the overall accuracy obtained is 73%.

3. Implementation

3.1 Approach

In this project, my approach is to develop a reliable and accurate system for face detection and emotion recognition, Where I have used Convolutional Neural Network (CNN) for the model training . After training the model, it is integrated with OpenCV. In OpenCV the face can be detected and the emotions can be displayed. Additionally, recognized emotions can be heard.

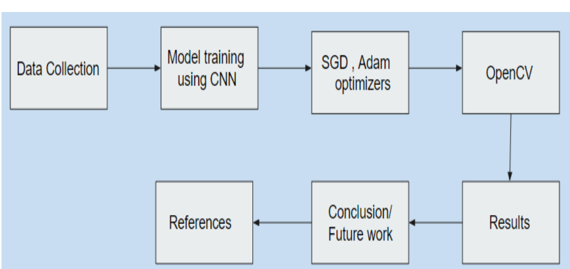

Fig 1. Approach

3.2 Data Collection

The data set that can be used in this project is taken from the public resource called Kaggle. This dataset has both train and validation data. The training data has seen different classes such as happy, sad, angry, disgust, neutral, surprise, fear. Similarly, the validation has seven different classes, they are happy, sad, angry, disgust, neutral, surprise, fear [4].

3.3 Convolutional Neural Network (CNN)

In face detection, CNN will analyze images by progressively learning intricate facial features through convolutional layers. These networks enable automatic feature extraction and localization, allowing them to identify faces within images using learned patterns, resulting in accurate and efficient detection.

CNN plays a pivotal role in emotion recognition by processing facial expressions through layers that capture features like eyes, nose, ears, forehead, mouth. These networks learn to differentiate emotions, allowing them to categorize faces into emotional states with a high degree of precision.

Five layers are taken in the CNN for the model training, where those five features are considered as the features. Such as eyes, nose, ears, forehead, mouth. BatchNormalization, Rectified Linear Unit (ReLu), MaxPooling2D, and Dropout are used for not overfitting the model.

```
#1st CNN layer
model.add(Conv2D(64,(3,3),padding = 'same',input_shape = (48,48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.5))

#2nd CNN layer
model.add(Conv2D(128,(5,5),padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout (0.5))

#3rd CNN layer
model.add(Conv2D(128,(5,5),padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout (0.5))

#4th CNN layer
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

#5th CNN layer
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))
```

Fig 2. Approach

3.3.1 Stochastic Gradient Descent (SGD)

SGD is a widely used optimization algorithm in training neural networks, including CNNs. It is particularly useful when dealing with large datasets because it updates the model's parameters based on small random batches of data rather than the entire dataset.

SGD features are almost similar to the Adam Optimizer features, but there is one drawback that made Adam Optimizer better is at first time learning if it faces any misdetection it will ignore that completely. But in the facial detection process we shouldn’t miss a single thing.

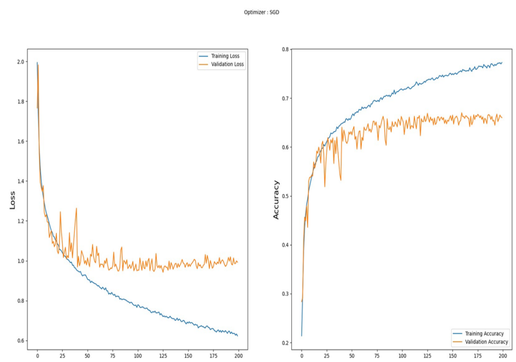

Fig 3. SGD Optimizer accuracy

3.3.2 Adaptive Moment Estimation (Adam)

Adam (Adaptive Moment Estimation) is a popular optimization algorithm that combines the advantages of both SGD and adaptive learning rate methods. It adapts the learning rate for each parameter individually based on past gradients and squared gradients.

Adam is mainly preferred because when it validates the dataset, there are less chances of getting errors. And if we are planning to take more epochs then it is better to use Adam than SGD.

Adam optimizer represents a significant advancement in the field of deep learning optimization. Its adaptability, efficiency, and ability to handle a wide range of neural network architectures make it a valuable tool for researchers and practitioners alike. Understanding the inner workings of Adam and its key features is essential for harnessing its power in training deep neural networks effectively.

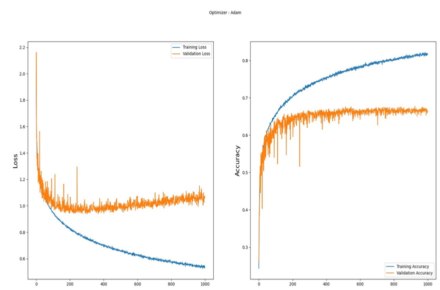

Fig 4. Adam Optimizer accuracy

3.2 OpenCV (Open-Source Computer Vision Library)

OpenCV (Open-Source Computer Vision Library) is a widely used open-source library for computer vision tasks, especially for face detection and emotion recognition.
OpenCV is an indispensable tool for integrating face detection and emotion recognition functionalities into applications.
Leveraging OpenCV's Haar cascades or deep learning models, it swiftly identifies faces in images or real-time video streams.
By incorporating its image processing capabilities and pre-trained models, OpenCV empowers people to accurately recognize facial expressions and emotions, making it a vital component for creating interactive and emotionally-aware systems.

4. Results

4.1 Expected Results

Fig 5. Expected Results

4.2 Obtained Results

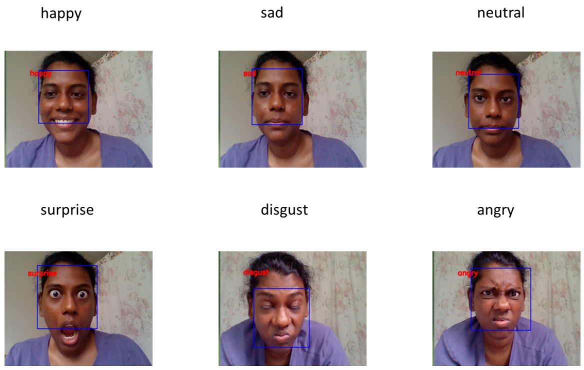

Fig 6. Obtained Results

4.3 Accuracy Results

Optimizers	Accuracy	Epochs
SGD	0.772	120
Adam	0.816	1000

Table 1. Results of both the optimizers

5. Conclusion

In conclusion, the choice between Stochastic Gradient Descent (SGD) and Adam optimization for training Convolutional Neural Networks (CNNs) is a critical decision that significantly impacts the model's performance. While both optimizers contribute to efficient convergence and parameter updates, my experimentation demonstrates that Adam consistently outperforms SGD in terms of accuracy. In scenarios where accuracy is important, my findings strongly recommend the adoption of Adam as the preferred optimization algorithm. The performance gains achieved by leveraging Adam's adaptability, efficient memory usage, and hyperparameter optimization underline its suitability for pushing the boundaries of accuracy in CNN.

6. Future Work

Collecting the real time images of a person, in order to obtain better accuracy. It is always limited to learning only the six-basic emotion plus neutral. It conflicts with what is present in everyday life, which has emotions that are more complex. The future work is to build larger databases and create powerful deep learning architectures to recognize all basic and secondary emotions.

Moving from 2D images, 3D facial analysis could provide more detailed insights into emotions by considering depth, which could improve accuracy and realism.

Future implementations in the field of facial detection and emotion recognition could involve advancements in technology, increased integration into various applications, and addressing ethical considerations.

Acknowledgements

Assistance in this project was provided by Dr. Jiayin Wang, Department of Computer Science at Montclair State University with assisted materials and guidance through this project.

7. References

1. Song Z (2021) Facial Expression Emotion Recognition Model Integrating Philosophy and Machine Learning Theory. *Front. Psychol.* 12:759485. doi: 10.3389/fpsyg.2021.759485

2. Wafa Mellouk, Wahida Handouzi (2020) Facial emotion recognition using deep learning: review and insights, The 2nd International Workshop on the Future of Internet of Everything (FIoE) August 9-12, 2020, Leuven, Belgium

3. Ko BC. A Brief Review of Facial Emotion Recognition Based on Visual Information, Sensors (Basel). 2018 Jan 30; 18(2):401. Doi: 10.3390/s18020 PMID: 29385749; PMCID: PMC5856145

4.https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset

5.Mehendale, N. Facial emotion recognition using convolutional neural networks (FERC). *SN Appl. Sci.*2,446(2020).

6.https://doi.org/10.1007/s42452-020-22341

7.https://medium.com/analytics-vidhya/facial-emotion-recognition-hands-on-guide-8e23f3d0025f

8.https://www.mdpi.com/2078-2489/13/6/26

9.https://edps.europa.eu/system/files/2021-05/21-05-26_techdispatch-facial-emotion-recognition_ref_en.pdf