The North African Journal of Scientific Publishing (NAJSP)
مجلة شمال إفريقيا للنشر العلمي (NAJSP)
EISSN: 2959-4820
Volume 2, Issue 2, April – June 2024, Page No: 60-72
Website: https://najsp.com/index.php/home/index
SJIFactor 2023: 3.733 0.63 :2023 (AIF) معامل التأثير العربي ISI 2023: 0.383

Facial Expression Recognition Using Convolutional Neural Networks and Stochastic Gradient Descent Optimization Algorithms

Omar Balola Ali [1*], Abdulhafid Bughari [2], Mohammed Hameed Bubakr [3], Mohammed Alfateh Abdalmonem [4]
[1,2,3] Computer Engineering, Faculty of Science Engineering, Bright Star University – El-Brega, Libya
[4] Faculty of Computer Science and Information Technology, Omdurman Islamic University –Sudan

*Corresponding author: omar.balula@bsu.edu.ly

Received: February 28, 2024	Accepted: April 03, 2024	Published: May 24, 2024

Abstract:
This paper presents the design of a Facial Expression Recognition (FER) system using Deep Convolutional Neural Networks (DCNNs) to accurately identify seven key human facial expressions. The DCNN module and FER system were trained and tested on various facial datasets, including JAFFE, KDEF, MUG, WSEFEP, ADFES, and TFEID. The experiments involved testing different models and architectures with varying numbers of convolutional layers, filter sizes, and epochs. Results from these experiments are based on 2982 images of faces on the seven-basic expression. The study also evaluated the performance of Stochastic Gradient Descent (SGD), Root Mean Squared Propagation (RMSprop), and Adaptive Moment Estimation (Adam), optimization algorithms on the DCNN architecture. Results showed that SGDM with an adaptive learning-rate achieved the highest validation accuracy of 98.35%, outperforming other algorithms. Additionally, the study found that RMSprop led to unstable training and lower accuracy, while Adam did not significantly improve accuracy with adaptive learning rate.

The research demonstrated that selecting the right combination of model elements led to improved accuracy and convergence time. The system achieved a recognition rate of 98.35% for the tested dataset using the DCNN algorithm, highlighting its effectiveness in facial expression recognition.

Keywords: Facial Expression Recognition (FER), Stochastic Gradient Descent (SGD), Deep Convolutional Neural Networks (DCNNs), Root Mean Squared Propagation (RMSprop).

Cite this article as: O. B. Ali et al, "Facial Expression Recognition Using Convolutional Neural Networks and Stochastic Gradient Descent Optimization Algorithms," *The North African Journal of Scientific Publishing (NAJSP),* vol. 2, no. 2, pp. 60–72, April – June, 2024.

التعرف على تعابير الوجه باستخدام الشبكات العصبية الالتفافية العميقة وخوارزميات تحسين الانحدار التدرجي العشوائي

عمر بلولة علي[1*]، عبدالحفيظ بخاري[2]، محمد حميد بوبكر[3]، محمد الفاتح عبدالمنعم[4]
[1،2،3] قسم هندسة الحاسوب، كلية العلوم الهندسية، جامعة النجم الساطع، البريقة، ليبيا
[4] كلية الدراسات العليا، جامعة أم درمان ألإسلامية، السودان

الملخص
يقدم هذا البحث تصميم نظام التعرف على تعبيرات الوجه (FER) باستخدام الشبكات العصبية الالتفافية العميقة (DCNNS) لتحديد سبع تعبيرات رئيسية للوجه البشري. تم تطوير نموذج DCNN ونظام FER باستخدام بيئة MATLAB واختبارها على مجموعات بيانات المختلفة لصور الوجه، بما في ذلك Jaffe وKDEF وMug وWsefep وADFES وTfeid. تضمنت التجارب اختبار نماذج وهيكلية مختلفة بأعداد متفاوتة من الطبقات الالتفافية، وأحجام المرشحات، والدورات. استندت نتائج هذه التجارب إلى 2982 صورة للوجوه على التعبيرات السبع الأساسية. تم تقييم الدراسة أيضًا باستخدام خوارزميات: أداء نزول التدرج العشوائي (SGD)، والانتشار التربيعي للجذر (RMSprop)، وتقدير اللحظة التكيفية (Adam)، وخوارزميات التحسين على بنية DCNN. أظهرت النتائج أن SGDM مع معدل التعلم التكيفي حققت أعلى دقة للتحقق من الصحة بنسبة 98.35 ٪، تفوقت على الخوارزميات الأخرى. بالإضافة إلى ذلك، وجدت الدراسة أن RMSprop أدت إلى تدريب غير مستقر ودقة أقل، في حين أن (Adam) لم يحسن بشكل كبير من الدقة مع معدل التعلم التكيفي.
أظهر البحث أن اختيار المزيج الصحيح من عناصر النموذج أدى إلى تحسين الدقة ووقت التقارب. حقق النظام معدل اعتراف قدره 98.35 ٪ لمجموعة البيانات التي تم اختبارها باستخدام خوارزمية DCNN، مما يبرز فعاليتها في التعرف على تعبير الوجه.

الكلمات المفتاحية: التعرف على تعبيرات الوجه (*FER*)، والنسب التدرج العشوائي (*SGD*)، والشبكات العصبية التلافيفية العميقة (*DCNNs*)، وانتشار متوسط الجذر التربيعي (*RMSprop*).

Introduction

Facial Expression Recognition (FER) plays a crucial role in various fields, including human-computer interaction, affective computing, and psychological research. The ability to accurately identify and interpret human facial expressions has significant implications for applications such as emotion recognition, behavior analysis, and mental health assessment. In recent years, Deep Convolutional Neural Networks (DCNNs) have emerged as powerful tools for FER, demonstrating remarkable capabilities in learning and extracting complex features from facial images.

Facial expression recognition technology analyzes a person's facial expressions to determine their emotional state and intentions. By detecting and interpreting facial expressions, computer systems can gather valuable information about a person's emotional state and intentions [1], Automatic facial expression recognition by computers has facilitated the realization of numerous new applications. Traditional methods for facial expression recognition have employed techniques such as Gabor wavelet and SVM [2], or geometric feature extraction with manual intervention [3]. However, these methods often yield suboptimal results due to their reliance on human involvement.

Currently, deep learning approaches, particularly convolutional neural networks (CNNs), have shown superior performance in facial expression recognition. Researchers, like Tang [4], have combined CNNs with SVM for expression recognition, surpassing the performance of traditional methods. This shift to deep learning has significantly improved the accuracy and effectiveness of facial expression recognition. This paper aims to explore the effectiveness of DCNNs in FER and evaluate the impact of different optimization algorithms on model performance. The study involves the development and testing of a DCNN-based FER system using various datasets and model architectures. Additionally, the research seeks to identify the most suitable optimization algorithm for enhancing the accuracy and convergence of the FER model.

By investigating the interplay between DCNNs, optimization algorithms, and FER, this study aims to contribute valuable insights to the field of computer vision and deep learning. The findings have the potential to advance the development of robust and efficient FER systems, with implications for diverse domains such as human-computer interaction, healthcare, and affective computing.

The paper is organized as follows: Section 2 describes the methodology, section 3 describes experiments and results, section 4 the conclusion is presented in section 5.

Related work

Most of the studies on facial expression recognition used the whole face area, whereas a few of them divided the face into several blocks and extract features from these blocks. In [5], the face area of every image was equally divided into 6*7 patches and then the LBP features were extracted from these empirically weighted patches to represent the facial expressions. After that, SVM was applied to classify facial expression using the LBP features. However, this method suffers from fixed region size and positions. As a result, in [6], Shan et al. proposed boosting LBP features to solve these problems. The boosting LBP features were obtained by scaling sub-window over face images and boosted by AdaBoost. Lin et al. [6] proposed an approach to learn the effective patches statistically. In [6], the face area was divided into 8*8 patches and a multi-task sparse learning method was applied to learn the active facial patches. The experiment result showed that the active patches were around eyes, nose and mouth, which confirmed the discovery in psychology. Moreover, three different scale sizes

were used in [7]. Different from [5],[6],[8], Happy et al. [9] selected 19 active patches from face area, which was supported by [6].

Convolutional Neural Network is a unique method. It combines segmentation, feature extraction and classification in one processing module. Most of CNN design is derived from LeNet-5 which is a neural network architecture for handwritten and machine-printed character recognition proposed by Yann LeCun, Leon Bottou, Yosuha Bengio and Patrick Haffner in 1990's. LeNet-5 consists of 7 layers that is formed by 4 feature extraction layers and 3 layers of multilayer perceptron (MLP). Feature extraction layer consists of convolution and sub sampling layers. Convolution layer removes noise and detect lines, borders or corners of an image. In sub sampling layer, it reduces the resolution of an image to prevent image distortions. CNN has built-in invariance as compared to typical neural network (MLP). CNN is superior in producing high accuracy in identification process although the algorithm is complex. In training the network, LeNet-5 applies Stochastic Diagonal Levenberg Marquadt (SDLM) learning algorithm. Unlike other neural network, when other complex database is applied, the system has to go through minimal redesigning process.**خطأ! لم يتم العثور على مصدر المرجع.**

The provided text outlines popular optimization algorithms used in training neural networks, including Stochastic Gradient Descent (SGD), RMSprop, and Adaptive Moment Estimation (Adam). SGD is described as an iterative algorithm that updates model parameters based on the gradients of the loss function, utilizing random mini batches to achieve faster convergence. RMSprop is highlighted for its ability to adapt the learning rate for each parameter, addressing the limitations of SGD and promoting faster and more reliable convergence. Adam, on the other hand, combines the concepts of both SGD and RMSprop, effectively handling sparse gradients and noisy objectives. It's emphasized that the choice of optimization algorithm depends on specific task requirements, dataset characteristics, and neural network attributes. While SGD, RMSprop, and Adam are popular choices, other optimization algorithms are available, each with unique advantages and considerations.

Gradient Descent Optimization Algorithms

Gradient descent is an optimization method for finding the minimum of a function. It is commonly used in deep learning models to update the weights of the neural network through backpropagation.

- Stochastic Gradient Descent (SGD)

The vanilla gradient descent updates the current weight w using the current gradient ∂L/∂w multiplied by some factor called the learning rate, α.

$$w_{t+1} = w_t - \alpha \frac{\partial L}{\partial w_t} \tag{1}$$

- RMSprop

Root Mean Squared Propagation (RMSprop) is a gradient-based optimization technique proposed by Geoffrey Hinton at his Neural Networks Coursera course **.خطأ! لم يتم العثور على مصدر المرجع**, It uses a moving average of squared gradients to normalize the gradient itself. That has an effect of balancing the step size-decrease the step for large gradient to avoid exploding and increase the step for small gradient to avoid vanishing. **.خطأ! لم يتم العثور على مصدر المرجع**.

It is defined as:

$$D[\phi]_\tau = DR.D[\phi]_{\tau-1} + (1 - DR)\phi_\tau \tag{2}$$

$$w^{\tau+1} = w^t - \frac{\eta}{\sqrt{D[\nabla E(w^\tau)^2]_\tau + \epsilon}}.\nabla E(w^\tau) \tag{3}$$

Where $D[\nabla E(w\tau)2]_\tau$ is the moving average of squared gradients at iteration τ; DR (decay rate) is a hyper parameter whose usual values are: 0.9, 0.99, or 0.999 corresp onding to averaging lengths of 10, 100, and 1000 parameter updates, respectively **.خطأ! لم يتم العثور على مصدر المرجع, خطأ! لم يتم العثور على مصدر المرجع.,خطأ! لم يتم العثور على مصدر المرجع**.

- Adam

Adaptive Moment Estimation (Adam) is an alternative method that calculates the adaptive learning rates for each parameter. Adam combines "the ability of Adagrad to deal with sparse gradients, and the ability of RMSprop to deal with non-stationary objectives" [15] by keeping track of both an exponentially decaying average of past square gradients $v\tau$, and an exponentially decaying average

of past gradients $m\tau$. The first element corresponds to the second moment of the gradients (variance). The second element represents the first moment of the gradients (mean). $v\tau$ and $m\tau$ values correspond to D [ϕ 2] τ and D[ϕ]τ, respectively [16].

As it is mentioned in [15], $v\tau$ and $m\tau$ are initialized as vectors of zeros. This causes the moment estimates to be biased towards zero, particularly when the decay rates are small and during the initial time steps. They counteract these biases in the following way:

$$\hat{m}_\tau = \frac{m_\tau}{1 - \beta_1^\tau} \tag{4}$$

$$\hat{v}_\tau = \frac{v_\tau}{1 - \beta_2^\tau} \tag{5}$$

Where $\beta 1$ and $\beta 2$ are exponential decay rates for the moment estimates; and $\hat{m\tau}$ and $\hat{v\tau}$ are the bias-corrected first moment estimate and second raw moment estimate, respectively.

Material and methods

In this section, we focus on the methodologies of the facial expression recognition problem. The stages of facial expression recognition to train the data set and recognition modules will be described. The overview of the FER system is illustrated in Figure. 1. The FER system includes the major stages such as image acquisition, image preprocessing, feature extraction and classification.

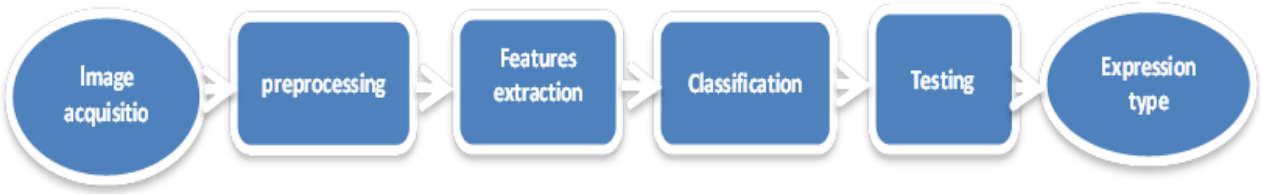

Figure 1: General facial expression recognition framework.

Data Collection

This section provides an overview of the datasets used for training the model in order to prevent bias towards any specific dataset. The following six datasets were collected from different sources:

KDEF Database

The Karolinska Directed Emotional Faces (KDEF) database consists of 4900 pictures capturing human facial expressions of emotion. It includes 70 individuals, each displaying seven different expressions. Each expression was photographed twice from five different angles [17]. Figure. 2 displays a sample of facial expressions from the dataset.

Figure 2: Sample images from the KDEF dataset.

AFFE Database:

The Japanese Female Facial Expression (JAFFE) database comprises 213 photographs of 10 Japanese female models exhibiting seven facial expressions, including six basic facial expressions and one neutral expression [18]. Figure. 3 displays a sample of facial expressions from the dataset.

Figure. 3: Sample images from the JAFFE dataset.

MUG Database:

The MUG database contains numerous sequences featuring an ample number of subjects for the development and evaluation of facial expression recognition systems. It focuses on posed expressions of the six basic emotions and exhibits different emotional reactions from subjects in response to the same stimuli [19]. Figure. 4 displays a sample of facial expressions from the dataset.

Figure 4: Sample images from the MUG dataset.

WSEFEP Database:

The Warsaw Set of Emotional Facial Expression Pictures (WSEFEP) consists of 210 high-quality pictures capturing 30 individuals. This dataset is widely recognized within the scientific community and frequently used as research material [20]. Figure. 5 displays a sample of facial expressions from the dataset.

Figure 5: Sample images from the WSEFEP dataset.

ADFES Database:

The ADFES database is the first standardized set of dynamic filmed expressions. Studies indicate that emotions displayed in the ADFES dataset are highly recognizable. The direction of expression, achieved through head-turning, influences the perceived cause of the emotion but does not affect recognition [21]. Figure. 6 displays a sample of facial expressions from the dataset.

Figure 6: Sample images from the ADFES dataset.

TFEID Database:

The TFEID database comprises 7200 stimuli obtained from 40 models, including 20 males. Each model presents eight facial expressions: neutral, anger, contempt, disgust, fear, happiness, sadness, and surprise [22]. Figure. 7 displays a sample of facial expressions from the dataset.

Figure 7: Sample images from the TFEID dataset.

Preprocessing

In the preprocessing stage of the image datasets, two main steps were performed: face detection and data modification, face detection involved using a cascade object detector, specifically the Viola-Jones algorithm, to detect faces in the dataset images. The detected faces were then cropped and extracted for further processing.

After face detection, the extracted face images underwent several modifications. First, they were converted to grayscale since facial expressions can be adequately identified using grayscale images. Next, the images were resized to dimensions of 100x100.

To ensure the data dimensions were on a similar scale, a technique called Zero-Center normalization was applied. This involved dividing each dimension (channel) by its standard deviation after centering the data around zero by subtracting the mean. The Zero-Center normalization equation is as follows:

$$\hat{x} = \frac{x - \mu_x}{\sigma_x} \quad \textbf{(6)}$$

where $\hat{x}$ represents the normalized feature vector, x is the original feature vector, μ_x denotes the mean of x, and σ_x is the standard deviation of x.

Features extraction

Feature extraction with Convolutional Neural Networks (CNNs) involves converting pixel data into a higher-level representation that captures various visual attributes of the input, such as shape, motion, color, texture, and spatial configuration. CNNs achieve this by utilizing convolutional kernels, which are convolved over the entire image. This convolution process generates feature maps that highlight the presence and location of specific features within the image. These feature maps, shown in Figure 8, serve as input for subsequent layers in the network. By processing these feature maps through additional layers, CNNs can learn hierarchical representations of the input data. This hierarchical learning enables them to better understand and interpret visual content.

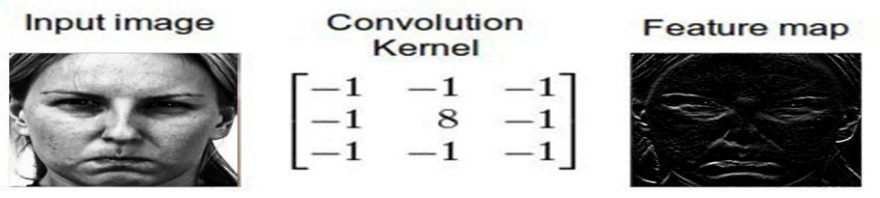

Figure 8: Feature map.

Classification

Facial expression recognition research deals with a multi-classification challenge. To develop a model capable of accurately classifying facial expressions, Convolutional Neural Networks (CNNs) have the ability to predict class labels for classification tasks or real values for regression tasks. In essence, the

classification objective is accomplished by combining a fully connected layer with the other algorithms [23]. CNN architectures for facial expression recognition as shown in Figure. 9, typically consist of several key components. Convolutional Layers: These layers apply a set of learnable filters to the input image, enabling the network to extract relevant features. Each filter performs convolutions across the input image, producing feature maps that highlight different aspects, such as edges, textures, or facial landmarks. Pooling Layers: These layers down sample the feature maps, reducing their spatial dimensions while preserving their essential information. Common pooling operations include max pooling or average pooling, which capture the most prominent features within each region. Fully Connected Layers: These layers connect all the neurons from the previous layers to the subsequent layers, allowing for complex combinations of features. In facial expression recognition, fully connected layers are typically used at the end of the network to perform the final classification based on the extracted features.

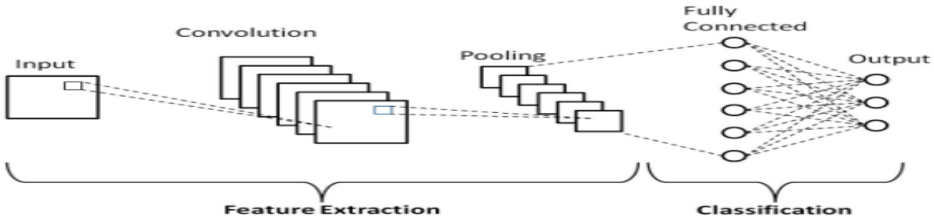

Figure 9: the architecture of basic Convolutional Neural Network.

Results and discussion

The effectiveness of optimization algorithms plays a crucial role in training deep neural networks for Facial Expression Recognition (FER) tasks. In this section, we present the results of our evaluation, which focuses on comparing the performance of three widely used optimization algorithms: Stochastic Gradient Descent (SGD), RMSprop, and Adaptive Moment Estimation (Adam).

The objective of this evaluation is to understand how these optimization algorithms impact the training process and the overall performance of the Deep Convolutional Neural Network (DCNN) architecture for facial expression recognition. By systematically comparing and analyzing the results, we aim to provide insights into the strengths and limitations of each algorithm in the context of FER.

These findings have significant implications for the design and selection of optimization algorithms for FER tasks. By understanding the strengths and weaknesses of each algorithm, researchers and practitioners can make informed decisions when training DCNN models for facial expression recognition applications.

In the subsequent sections, we present and discuss the detailed results of our evaluation, providing quantitative performance metrics, visualizations, and analysis. These findings contribute to the growing body of knowledge in optimizing deep neural networks for FER and provide valuable insights for further improvements in this domain.

Experiments with image size 100x100

In these experiments we used 100x100 images size with different numbers of convolutional layers, one, two and three. Also, with filter size, 2, 6 and 10 and numbers of epoch, 20, 35 and 50. Analyzing results as shown in tables 1, table 2 and table 3 show the improvement at these experiments where the classifying accuracy are 93.46%, 94.6% and 93.23% respectively.

Table 1 The result of 100x100 image size and one convolutional layer.

Conv2d layers	Filter size	Epoch	Accuracy rate %
1	2	20	88.4615
		35	91.7308
		50	91.5385
	6	20	89.8077
		35	91.5385
		50	91.5385
	10	20	89.6156
		35	**93.4615**
		50	92.3077

The highest achieved accuracy is 93.4615 using one convolutional Layer with filter size of 10.

Table 2: The result of 100x100 images size two of convolutional layers.

Conv2d layers	Filter size	Epoch	Accuracy rate %
2	2	20	88.0769
		35	94.0385
		50	92.5000
	6	20	90.5769
		35	92.6923
		50	93.8462
	10	20	91.7308
		35	93.0769
		50	**94.6154**

The highest achieved accuracy is 94.6154 using two convolutional Layers with filter size of 10.

Table 3: The result of 100x100 image size and three convolutional layers.

Conv2d layers	Filter size	Epoch	Accuracy rate %
3	2	20	91.3462
		35	95
		50	94.2308
	6	20	92.6923
		35	93.0769
		50	92.1154
	10	20	90
		35	93.4615
		50	93.8462

The highest achieved accuracy is 95 using three convolutional Layer with filter size of 2. The highest accuracy result we achieve is 95 as shown table 3, using 3 convolutional layers and filter size of 6.

Table 4: The highest accuracy results for each experiment on 100x100 image size.

Experiment.no	Conv2d layers	Filter size	Epoch	Accuracy rate %
1	1	10	35	**93.46**
2	2	10	50	**94.61**
3	3	2	35	**95**

Hyper parameters and Learning rate

In this section, we present the results of our experiments, which were conducted to evaluate the impact of hyperparameters and learning rate on the performance of the Deep Convolutional Neural Network (DCNN) architecture. The DCNN architecture utilized in these experiments consists of 8, 16, and 32 filters in its convolutional layers, along with fully connected layers comprising 512 neurons each.

To explore the effect of different gradient descent optimization algorithms, we applied the experiments using three algorithms: Stochastic Gradient Descent with Momentum (SGDM), RMSprop, and Adaptive Moment Estimation (Adam). These optimization algorithms play a crucial role in training deep neural networks and can significantly affect the model's convergence and generalization capabilities.

Furthermore, we investigated the impact of varying the learning rate values. Initially, we set the learning rate values in the range of 0.01 to 0.00001 and observed their impact on the model's performance. Subsequently, we employed adaptive learning rate techniques to dynamically adjust the learning rate during the training process.

In the following sections, we describe the experimental setup, and present the outcomes and analyses of the experiments conducted with different hyper parameter configurations and learning rate variations.

Result using SGDM.

The results of using SGDM algorithm with consistent learning-rate (from 0.01 to 0.00001) and adaptive learning-rate with learning-rate drop by 0.1 every 5, 10, or 15 epochs are displayed in the are displayed in Table 5 and Table 6 respectively. Additionally, Figure 10 displays the corresponding results graphically.

Table 5: Accuracy of applying SDGM algorithms with Learning rate.

Learning rate	epochs	Accuracy %
0.01	16	92.56
0.00001	38	95.05
0.00001	455	**97.61**
0.00001	797	96.47

Table 6: Accuracy of applying SDGM algorithms with Adaptive Learning rate.

Adaptive Learning rate	epochs	Accuracy %
5	164	**98.35**
10	27	96.70
15	18	93.96

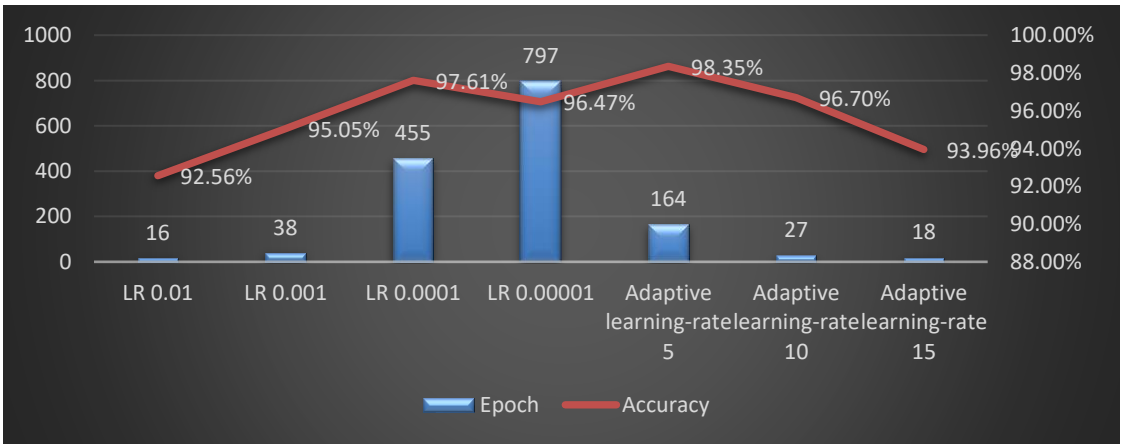

Figure 10: Accuracy of applying SDGM algorithms.

Lower learning rates required more time (epoch) to train. For instance, training with a learning rate of 0.00001 took 797 epochs, while a learning rate of 0.01 only took 16 epochs. The learning rate of 0.0001 achieved the highest validation accuracy of 97.61%. When using adaptive learning rate with the default moment value, training took 164 epochs and resulted in an accuracy of 98.35% Additionally, Figure 11 displays the corresponding results graphically.

Result using Adam.

The results of using Adam algorithm with consistent learning-rate (from 0.01 to 0.00001) and adaptive learning-rate with the typical values of the decay rate (0.9, 0.99, and 0.999) are displayed in Table 7 and Table 8 respectively.

Table 7: Accuracy of applying **Adam** algorithms with Learning rate**.**

Learning rate	epochs	Accuracy %
0.01	23	71.98
0.00001	21	93.41
0.00001	32	94.51
0.00001	161	95.06

Table 8: Accuracy of applying **Adam** algorithms with Adaptive Learning rate**.**

Adaptive Learning rate	epochs	Accuracy %
0.9	21	90.11
0.99	76	81.32
0.999	23	81.32

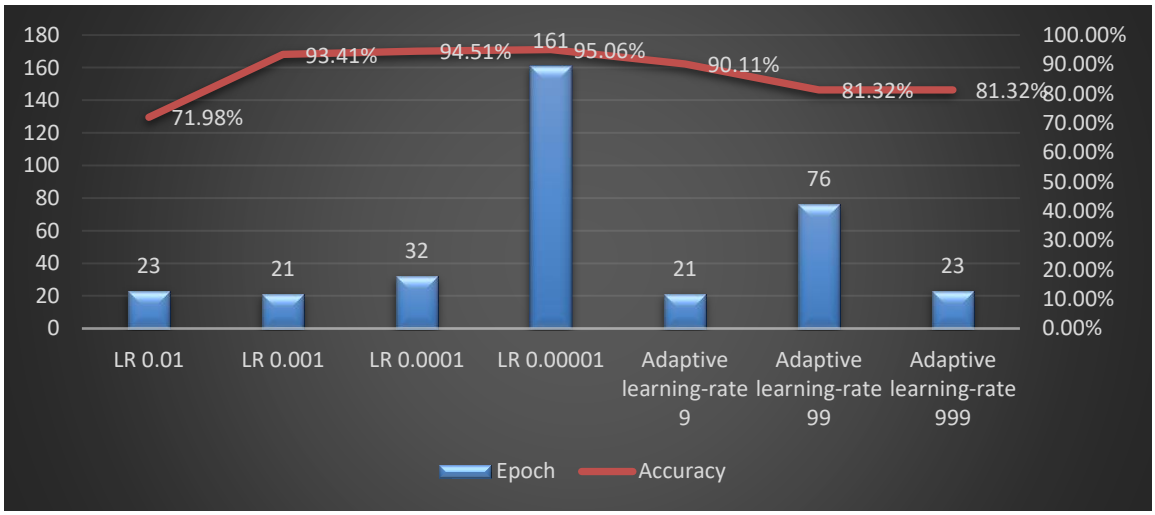

Figure 11: Accuracy of applying ADAM algorithms.

Lower learning rates yielded higher accuracy. For example, a learning rate of 0.00001 achieved an accuracy of 95.06% after 161 epochs. Adaptive learning rate did not provide any significant improvement in accuracy compared to a fixed learning rate.

Result using RMSprop.

The results of using RMSprop algorithm with consistent learning-rate (from 0.01 to 0.00001) and adaptive learning-rate with the typical values of the decay rate (0.9, 0.99, and 0.999) are displayed in are displayed in Table 9 and Table 10 respectively Additionally, Figure. 12 displays the corresponding results graphically.

Table 9: Accuracy of applying **RMSprop** algorithms with Learning rate**.**

Learning rate	epochs	Accuracy %
0.01	11	59.89
0.00001	22	91.76
0.00001	12	85.71
0.00001	72	95.06

Table 10: Accuracy of applying **RMSprop** algorithms with Adaptive Learning rate**.**

Adaptive Learning rate	epochs	Accuracy %
0.9	19	84.62
0.99	19	51.10
0.999	9	15.39

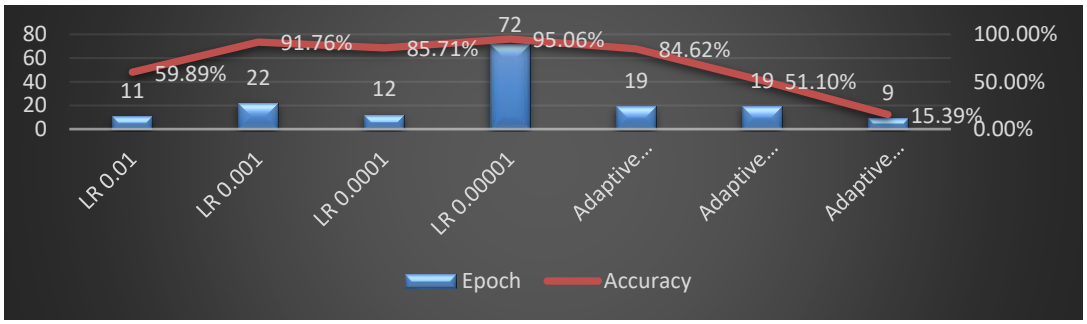

Figure 12: Accuracy of applying RMSprop algorithms.

Similar to Adam, lower learning rates led to higher accuracy. A learning rate of 0.00001 achieved an accuracy of 95.06% after 72 epochs. Adaptive learning rate did not result in any notable improvement in accuracy compared to using a fixed learning rate.

Table 13: The highest recognition rate achieved by the three aforementioned algorithms.

Techniques type	High accuracy
SGDM algorithm	98.35%.
Adam algorithm	95.06%
RMSprop algorithm	95.06

These results highlight the sensitivity of model performance to the choice of learning rate in the SGDM, ADAM, and RMSprop optimization algorithms. Lower learning rates tend to require more training epochs but can lead to higher accuracy, while adaptive learning rate techniques may not always yield superior results. Further analysis and experimentation are necessary to fully understand the behavior of these algorithms and their impact on the model's performance.

So, it is evident from the results that the RMSprop algorithm is not suitable for our model due to its tendency to result in unstable training and lower accuracy compared to other algorithms.

In conclusion, the SGDM (Stochastic Gradient Descent with Momentum) algorithm proves to be the most suitable and effective gradient descent optimization algorithm for our model. This is supported by the observation that the training progress curves for SGDM are smoother compared to Adam and RMSprop algorithms. The smoother curves indicate that SGDM achieves a more gradual and controlled convergence, resulting in better accuracy. Moreover, SGDM's momentum feature contributes to its effectiveness by allowing larger update values when the algorithm reaches a steady state.

Conclusion

This paper presents a comprehensive exploration of Facial Expression Recognition (FER) using Deep Convolutional Neural Networks (DCNNs) and various optimization algorithms. Through experiments on different datasets and model architectures, the study identified the most effective combination of elements for achieving high accuracy in recognizing facial expressions.

The research found that Stochastic Gradient Descent with adaptive learning-rate (SGDM) outperformed RMSprop and Adam optimization algorithms, achieving a recognition rate of 98.35% for the tested dataset. Additionally, the study highlighted the importance of selecting the right combination of model elements, such as the number of convolutional layers, filter sizes, and epochs, to improve accuracy and convergence time.

Overall, the findings demonstrate the efficacy of DCNNs in facial expression recognition and emphasize the significance of optimization algorithms in enhancing the performance of the model. The results provide valuable insights for the development of advanced FER systems and contribute to the ongoing research in the field of computer vision and deep learning.

References

[1] Y. Fu Guang. and Q. Lauding, "Facial Expression Recognition Based on Convolutional Neural Network Fusion SIFT Features of Mobile Virtual Reality", Wireless Communications and Mobile Computing, Volume 2023, Article ID 5763626, 2023.

[2] E. Sariyanidi, E. Granger, & J. Thiran "Automatic facial expression recognition using spatiotemporal Gabor features and supprt vector machines", IEEE International Conference on Image Processing (ICIP) (pp. 4001-4005), 2015.

[3] A. Lanitis, C. J. Taylor & T. F. Cootes, "Automatic interpretation and coding of face images using flexible models". IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 743-756. 1997.

[4] Y. Tang, "Facial expression recognition based on deep learning and support vector machine". 13th International Conference on Computer Science & Education (ICCSE) (pp. 123-127). IEEE. 2018.

[5] S. Zhu, C. Li, C. Change Loy, X. Tang, "Face alignment by coarse-to-fine shape searching", Proceedings of the IEEE conference on computer vision and pattern recognition, 4998–5006, 2015.

[6] C. Shan, S. Gong, PW. McCowan, " Facial expression recognition based on local binary patterns: a comprehensive study", Image Vision Compute 27(6):803–816, 2009.

[7] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, DN. Metaxas, "Learning active facial patches for expression analysis", IEEE conference on computer vision and pattern recognition (CVPR), p. 2562–9, 2012.

[8] L. Zhong, Q. Liu, P. Yang, J. Huang, DN. Metaxas, "Learning multiscale active facial patches for expression analysis", IEEE Trans Cybernet 45(8),1499–1510, 2015.

[9] P. Burkert, F. Trier, MZ. Afzal, A. Dengel, M. Liwicki, "Dexpression: deep convolutional neural network for expression recognition",1509.05371, 2015.

[10] S. Happy, A. Routray,"Automatic facial expression recognition using features of salient facial patches", IEEE Trans Affect Comput 6(1),1–12, 2015.

[11] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-Based Learning Applied to Document Recognition", Proceedings of the IEEE 1-46, 1998.

[12] C. Rodrigues, "Exploring the transfer learning aspect of deep neural networks in facial information processing", University of Manchester, p.23, 2015.

[13] S. Rifai, Y. Bengio, A. Courville, P. Vincent, M. Mirza, "Disentangling Factors of Variation for Facial Expression Recognition", Vol 7577. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33783-3_58, 2012.

[14] D. Kingma, and J. Ba., "Adam: A method for Stochastic Optimization", Published as a conference paper at ICLR 2015, https://doi.org/10.48550/arXiv.1412.6980, 2015.

[15] V. Sudha, "A fast and robust emotion recognition system for real-world mobile phone data", IEEE International Conference on Multimedia Expo Workshops (ICMEW), pp. 1–6, 2015.

[16] Y. Wu and Q. Ji, "Discriminative Deep Face Shape Model for Facial Point Detection," Int. J. Computer. Vision, vol. 113, no. 1, pp. 37–53, 2015.

[17] Javan der Schalk, S. T. Hawk, A. H. Fischer, & B. Doosje, "Moving faces, looking places: Validation of the Amsterdam Dynamic Facial Expression Set (ADFES)". Emotion, 11(4), 907–920. https://doi.org/10.1037/a0023853, 2011.

[18] M. Biehl, D. Matsumoto, P. Ekman, V. Hearn, K. Heider, T. Kudoh, V. Ton, "Matsumoto and Ekman's Japanese and Caucasian Facial Expressions of Emotion (JACFEE)", Journal of Nonverbal Behavior. 21. 3-21. 1997.
[19] R. Lienhart., A. Kuranov, and V. Pisarevsky, "Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection.", Proceedings of the 25th DAGM Symposium on Pattern Recognition. Magdeburg, Germany, 2003.
[20] Y. Andrew & S. Pawan, "Role of color in face recognition", Journal of Vision. 2. 10.1167/2.7.596. 2010.
[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks". Advances in Neural Information Processing Systems. Vol 25, 2012.
[22] J. Lia, D. Zhanga, J. Zhanga, J. Zhanga, T. Lia, Y. Xiaa, Q. Yana, and L. Xuna, "Facial Expression Recognition with Faster R-CNN", Published by Elsevier, 2017.
[23] N. Aifanti; Ch. Papachristou; A. Delopoulos, "The MUG facial expression database", IEEE, 1 – 4, 2010.