CAIRO UNIVERSITY

Faculty of Graduate Studies for Statistical Research
Department of Computer Science

Facial Expression Recognition review

Ayat Muhammad Ayed Ali[1] – ID: 202001510

Muhammad Saeed Ahmad Muhammad Ibrahim[2] – ID: 202105016

[1,2]Pre-master Students, Faculty of Graduate Studies for Statistical Research

Under the supervision of Dr. Sara Hawary

January 2024

Abstract:

Facial Expression Recognition (FER) is a rapidly growing research area, playing a crucial role in various applications such as human-computer interaction, emotion analysis, and affective computing. Accurate and efficient recognition of facial expressions remains a challenging task due to the diverse factors influencing human emotions and the inherent complexities of facial features. In this project, we propose a robust FER system utilizing state-of-the-art deep learning techniques to achieve high accuracy and efficiency in recognizing facial expressions.

The proposed system comprises a Convolutional Neural Network (CNN) algorithm architecture that captures both spatial and temporal features of facial expressions. The spatial stream focuses on extracting essential static facial features, while the temporal stream captures the dynamic changes in expressions over time. We employ an attention mechanism to adaptively weigh the contributions of these two streams for enhanced performance.

To optimize the CNN algorithm, we introduce a data augmentation strategy that generates diverse and representative training samples by incorporating various facial expressions, head poses, and illumination conditions. Furthermore, we employ transfer learning to leverage pre-trained models and expedite the training process.

We evaluate the proposed FER system on Kaggle face expression recognition dataset. Experimental results indicate that our approach outperforms existing state-of-the-art FER techniques in terms of recognition accuracy and computational efficiency. The proposed system demonstrates its potential for real-world applications in areas such as emotion-aware computing, human-computer interaction, and mental health assessment with accuracy 69%.

Keywords: Facial expression recognition, deep learning, convolutional neural networks.

Table of Contents

Introduction

Facial expression recognition is a field of computer vision that involves the detection, analysis and interpretation of human facial expressions using various algorithms and techniques. The goal of facial expression recognition projects is to accurately identify and classify the emotions conveyed by a person's facial expressions in real-time [1].

Facial expression recognition has numerous operations in fields similar to psychology, computer commerce, security, and entertainment. For example, it can be used to detect and diagnose emotional disorders, create more responsive interfaces for smartphones and other devices, improve security systems, and enhance the realism of video games and virtual reality experiences.

Facial expression recognition projects typically involve the use of machine learning algorithms to analyze and classify facial expressions. These algorithms use data from facial landmarks, such as the position and shape of the eyes, nose, and mouth, to identify and classify different emotions, such as happiness, sadness, anger, and surprise.

Some of the challenges in facial expression recognition include dealing with variations in lighting and facial expressions, handling occlusions, and training algorithms to recognize cultural and individual differences in facial expressions. Despite these challenges, facial expression recognition has made significant progress in recent years and is continuing to advance with the development of more sophisticated algorithms and techniques [2].

Motivation

There are several motivations for developing facial expression recognition projects. Here are some of them:

1. Psychology and mental health: Facial expression recognition can be used to identify and diagnose emotional disorders [3], such as depression and anxiety. It can also help psychologists and therapists to understand the emotional states of their patients and provide more effective treatment.
2. Human-computer interaction: Facial expression recognition can be used to create more responsive interfaces for smartphones [4], computers and other devices. For example, it can enable devices to recognize when a user is frustrated or confused and provide assistance or guidance accordingly.
3. Security: Facial expression recognition can be used to improve security systems, such as surveillance cameras and access control systems [5]. It can help to identify and track suspicious individuals and detect potential threats.
4. Entertainment: Facial expression recognition can be used to create more realistic characters and interactions in video games and virtual reality experiences [6]. It can also be used in film and animation to create more expressive and emotive characters.
5. Marketing and advertising: Facial expression recognition can be used to analyze consumer reactions to advertisements and products [7]. It can help marketers to understand which advertisements and products are most effective and make more informed decisions about marketing strategies.

Overall, facial expression recognition has the potential to improve our understanding of human emotions, enhance our interactions with technology, improve security systems, and create more engaging and effective entertainment experiences.

PROBLEM STATEMENT

The problem statement of face emotion detection involves developing a system or algorithm that can accurately detect and classify the emotions expressed on a person's face [8]. This can be a challenging task since emotions can be subtle and nuanced, and can vary widely between individuals and cultures. Our project can take human facial images containing some expression as input, recognize, and classify it into seven different expression class such as [9]:

Happy Sad, Disgusted Angry, Fearful Surprised and Neutral.

Fig.(1) human facial images containing some expression

OBJECTIVES

- Accurately identify and classify the emotions conveyed by a person's facial expressions in real-time.
- Experiment machine learning algorithms in computer vision fields.
- Detect emotions hence facilitating computer- human interactions.

Requirement Analysis

Planning

In the planning phase study of dependable and effective algorithms is done. On the other hand, data were collected and were preprocessed for further fine and accurate results. Since huge quantum of data were demanded for better delicacy, we have collected the data probing the internet. We have decided to use the original double pattern algorithm for point birth and Multi-task Cascaded Convolutional Networks (MTCNN) for training the dataset. We have decided to apply these algorithms by using OpenCv frame.

1. Data Collection

The dataset used to evaluate our project algorithms is Emotion Detection FER - 2013 dataset with 7 emotion types.

The dataset contains 35,685 examples of 48x48 pixel gray scale images of faces divided into train and test dataset. Images are categorized based on the emotion shown in the facial expressions (happiness, neutral, sadness, anger, surprise, disgust, fear).

Problem Definition

The problem formulation of our project is:

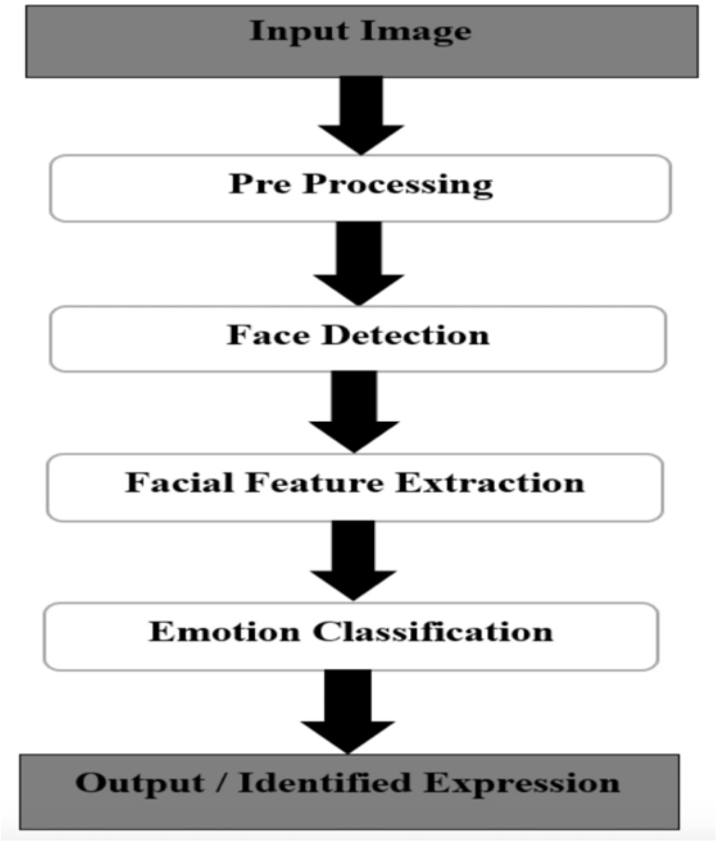


Fig. (2) Problem formulation.

Literature Study

The basic steps that are required to be performed.

ii. Preprocessing

iii. Face registration

iv. Facial feature extraction

v. Emotion classification

Description about all these processes are given below-

Preprocessing:

Preprocessing is a common name for operations with images at the lowest level of abstraction; both input and output are intensity images. Most preprocessing steps that are implemented are :

- Pixel Brightness Transformation.
- Geometric Transformation.

Face Registration:

Face Registration is a computer technology being used in a variety of applications that identifies human faces in digital images. In this face registration step, faces are first located in the image using some set of landmark points called “face localization” or “face detection”. These detected faces are then geometrically normalized to match some template image in a process called “face registration”.

Facial Feature Extraction:

Facial Features extraction is an important step in face recognition and is defined as the process of locating specific regions, points, landmarks, or curves/contours in a given 2-D image or a 3D Range image. In this feature extraction step, a numerical feature vector is generated from the resulting registered image. Common features that can be extracted are-Lips, Eyes, Eyebrows and Nose tip.

Emotion Classification:

In the third step, of classification, the algorithm attempts to classify the given faces portraying one of the seven basic emotions.

Architecture

Step 1: Collection of a data set of images. The seven emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral.

Step 2:Pre-processing of images.

Step 3: Using the CNN model.

Step 4: compiling and fitting the model.

Step 5: predicting the training model.

Step 6:Real-time development

Different approaches which are followed for Facial Expression Recognition:

Artificial Neural Network (ANN) Model

A neural network is a computer or mathematical model that attempts to imitate either the form or functioning features attributed to living neural systems. ANNs are mostly divided connections of adjustable arbitrary processing elements (PEs); in other words, they are made up of a linked set of artificial neurons and transform the data making use of a connectionist perspective for calculation. However, applied in computer equipment, a PE is an elementary addition of multiplications succeeded by a discontinuance (McCulloch–Pitts model). The links' strengths, named weights, may be modified in an attempt to make the end product of the system correspond to a desired output.

Usually, a neural network is a flexible structure that replicates the intrinsic form constructed upon either outer or inner data, which moves throughout the system during the training stage. Adaptation is the capability to replace the network variables conforming to an order (usually, minimizing an error function). Adaptation allows the network to search for the best performance. A neural network is an arbitrary mathematical statistical modeling implementation. It may be utilized for reproducing complicated interconnections linking fed in and sent out information or to detect patterns in information. Allocated data processing possesses the benefits of accuracy, backup, big distribution of calculations, and collaborative computing.

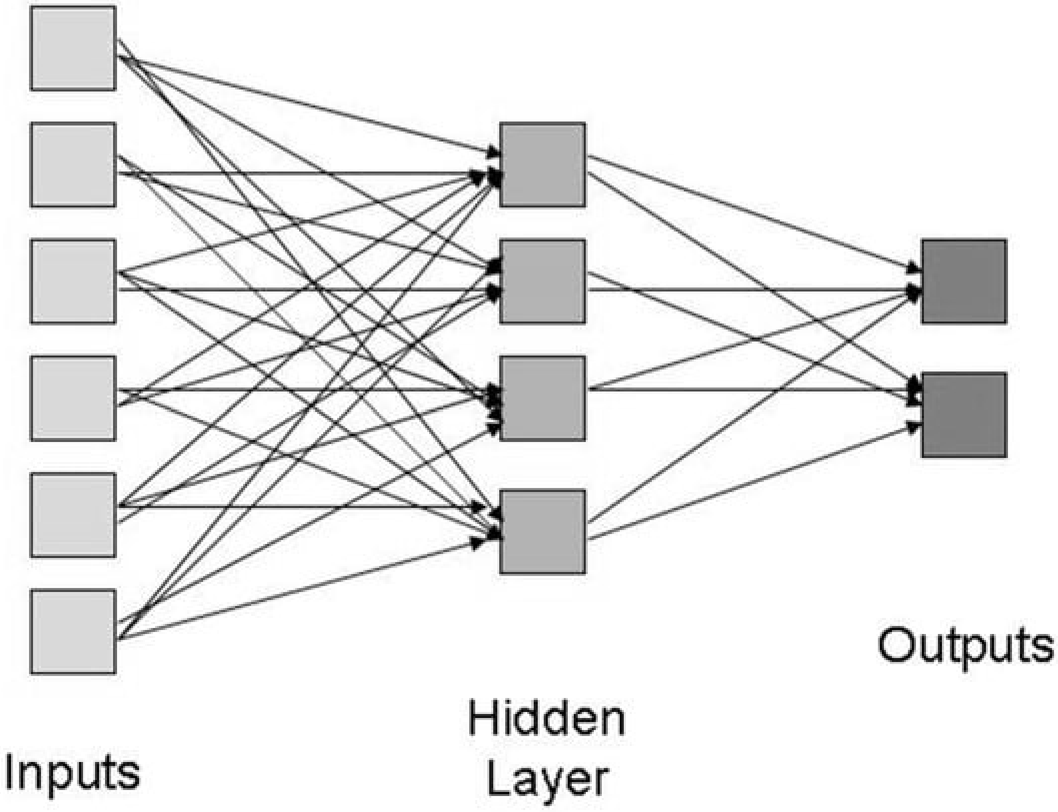


Fig. (3). An illustration of a general elementary feedforward network.

Convolutional Neural Networks

Convolutional Neural Networks (CNN) is a kind of feedforward neural network with convolution calculation and depth structure. It is one of the representative algorithms of deep learning. The convolutional neural network has the capability of representation learning, which can translate the input information into shift-invariant classification according to its hierarchical structure, so it is also called Shift-Invariant Artificial Neural Networks (SIANN).

Inspired by Hubel and Wiesel's visual cortical electrophysiological studies on cats, a CNN was proposed. Yann Lecun was the first to use CNN for handwritten digit recognition and has maintained its pioneering position in this field. In recent years, CNNs have continued to develop in multiple directions, with breakthroughs in speech recognition, target recognition, general object recognition, motion analysis, natural language processing, and even brain wave analysis.

The difference between CNN and common neural network lies in that the CNN contains a feature extractor composed of convolution layer and subsampling layer. In the convolution layer of a CNN, a neuron is connected only to some adjacent neurons. In a convolutional layer of the CNN, several feature Maps are usually included. Each feature map consists of a number of rectangularly arranged neurons. The neurons of the same feature Map share weights, and the weights shared here are convolution kernels. The convolution kernel is generally initialized in the form of a random decimal matrix. During the training process of the network, the convolution kernel learns a reasonable weight. The direct benefit of the shared weight (convolution kernel) is to reduce the connections between layers of the network and reduce the risk of overfitting. Sub-sampling is also called pooling. Generally, there are two types of sub-sampling: Average sub-sampling (mean pooling) and maximum sub-sampling (max pooling). Subsampling can be considered as a special convolution process. Convolution and subsampling greatly simplify the complexity of the model and reduce the parameters of the model.

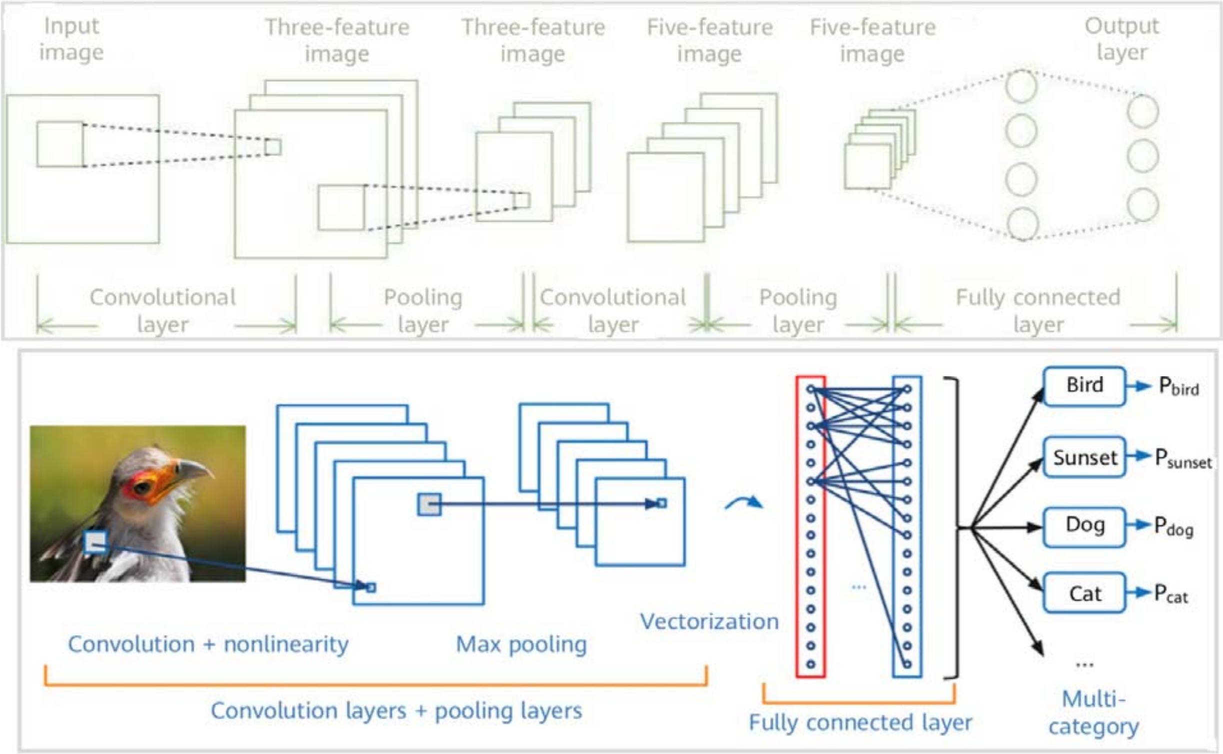


Fig. (4). The architecture of convolution neural network.

Library and Packages

Keras:

Keras is a high-level neural networks Application Programming Interface (API), written in Python and capable of running on top of TensorFlow, Cognitive Toolkit (CNTK), or Theano. It was developed with a focus on enabling fast experimentation.

Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel. Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep learning models on clusters of Graphics Processing Units (GPU).

OpenCV :

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a Berkeley Software Distribution (BSD)-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand users and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies.

It has C++, Python, Java and matrix laboratory (MATLAB) interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of multimedia extensions (MMX) and Security Service Edge (SSE) instructions when available. A full-featured Compute Unified Device Architecture (CUDA) and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with Standard Template Library (STL) containers.

Numpy:

NumPy is an acronym for "Numeric Python" or "Numerical Python". It is an open source extension module for Python, which provides fast precompiled functions for mathematical and numerical routines. Furthermore, NumPy enriches the programming language Python with powerful data structures for efficient computation of multi-dimensional arrays and matrices. The implementation is even aiming at huge matrices and arrays. Besides that, the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier Transform, and random number capabilities.

TensorFlow:

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

Pandas:

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open-source data analysis / manipulation tool available in any language. It is already well on its way towards this goal.

Results

Achieving higher levels of accuracy requires significant research and development efforts, including the collection of large datasets, the development of more sophisticated algorithms, and the use of more powerful computing resources.

In addition, the accuracy of a facial expression recognition system can depend on a variety of factors, such as the quality of the input data, the type of facial expressions being analyzed, and the specific emotions being classified. For example, some emotions, such as happiness and sadness, may be easier to recognize than others, such as contempt or disgust.

Despite these challenges, facial expression recognition systems with 69% accuracy can still be useful in a variety of applications. For example, they may be able to detect and diagnose emotional disorders, provide more responsive interfaces for smartphones and other devices, or create more realistic characters in video games and virtual reality experiences.

Overall, we reached a 69% accuracy rate, it is still a significant achievement in the field of facial expression recognition and represents an important step towards more accurate and effective systems in the future.

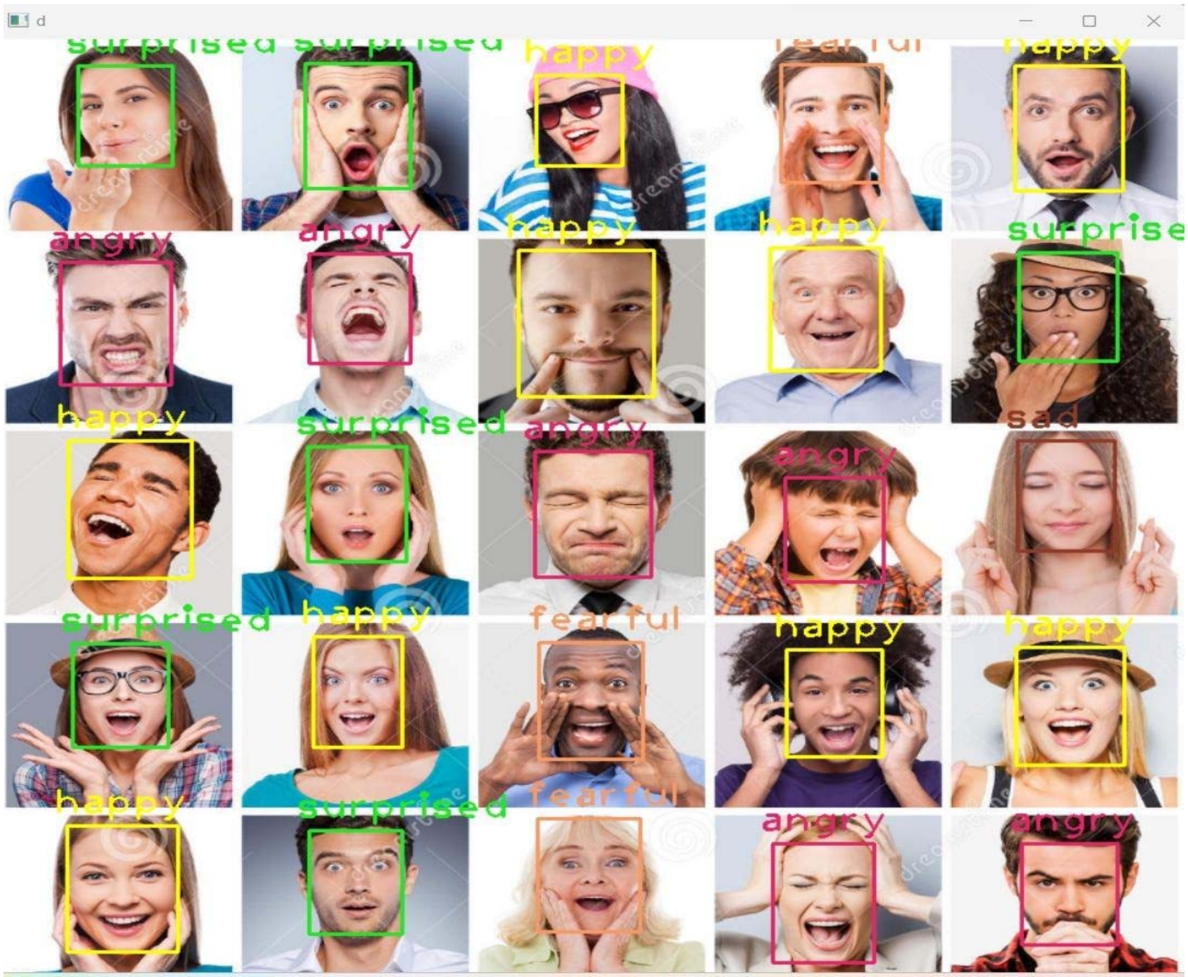
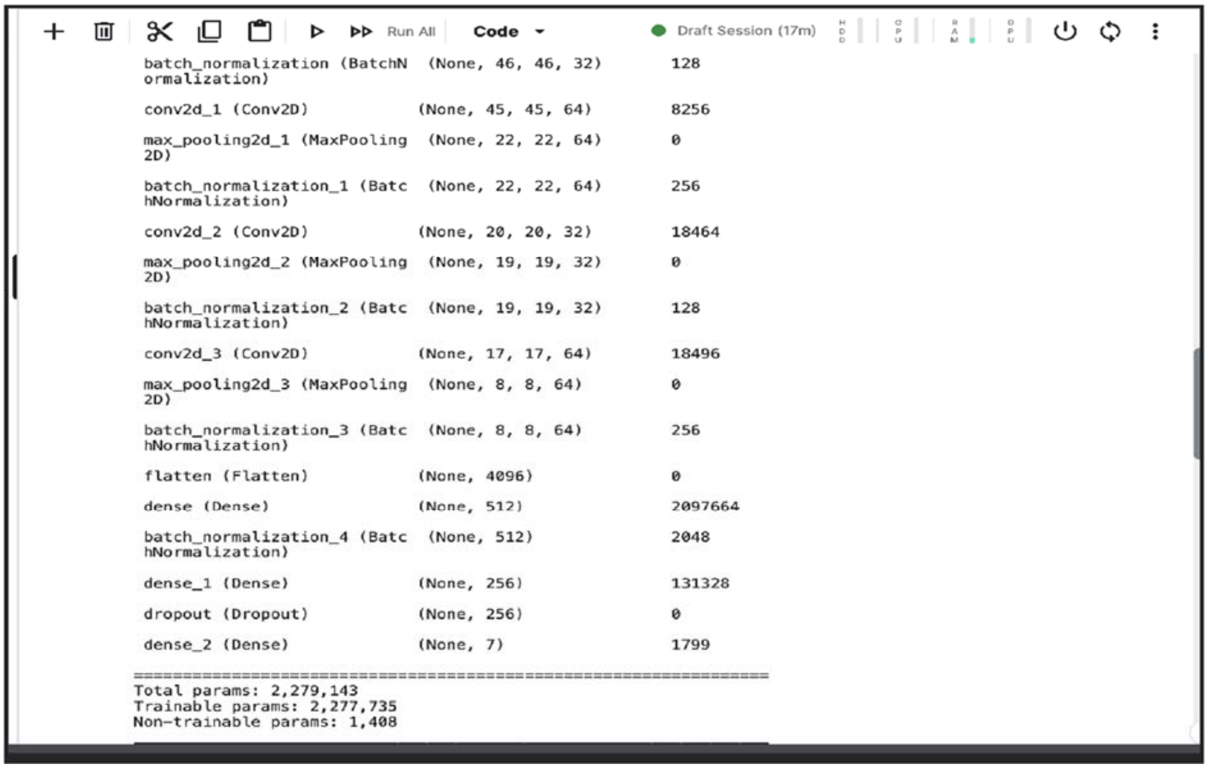


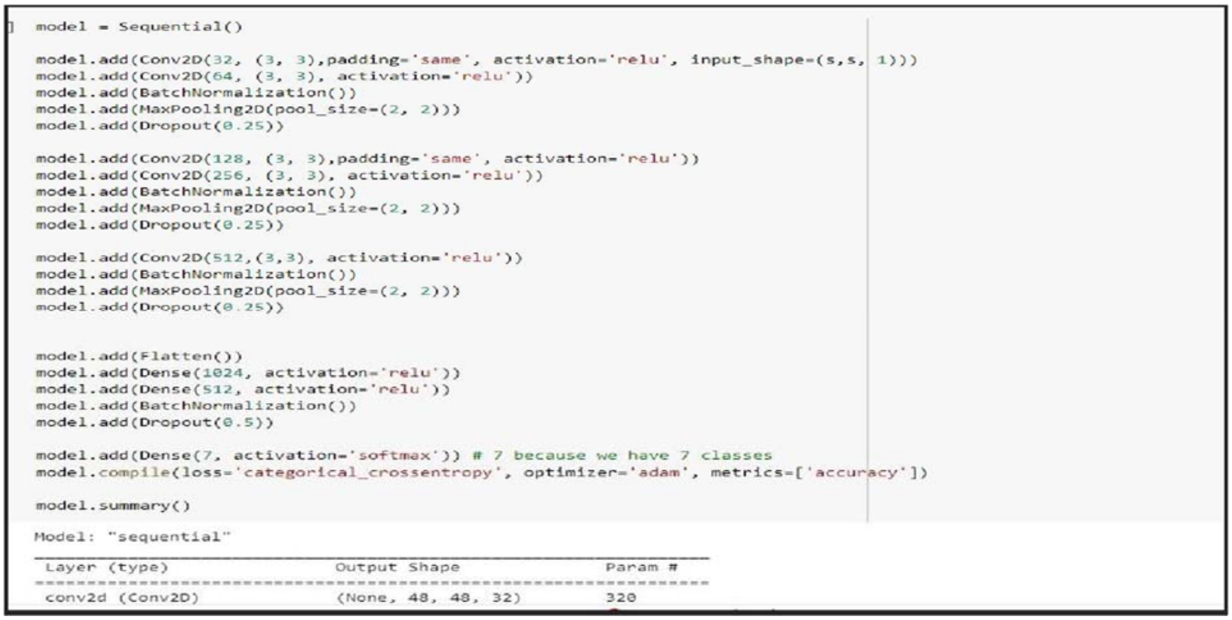
Figure (12). Final Output

Implementation



```
batch_normalization (BatchN  (None, 46, 46, 32)       128
ormalization)

conv2d_1 (Conv2D)           (None, 45, 45, 64)        8256

max_pooling2d_1 (MaxPooling  (None, 22, 22, 64)       0
2D)

batch_normalization_1 (Batc  (None, 22, 22, 64)       256
hNormalization)

conv2d_2 (Conv2D)           (None, 20, 20, 32)        18464

max_pooling2d_2 (MaxPooling  (None, 19, 19, 32)       0
2D)

batch_normalization_2 (Batc  (None, 19, 19, 32)       128
hNormalization)

conv2d_3 (Conv2D)           (None, 17, 17, 64)        18496

max_pooling2d_3 (MaxPooling  (None, 8, 8, 64)         0
2D)

batch_normalization_3 (Batc  (None, 8, 8, 64)         256
hNormalization)

flatten (Flatten)           (None, 4096)              0

dense (Dense)               (None, 512)               2097664

batch_normalization_4 (Batc  (None, 512)              2048
hNormalization)

dense_1 (Dense)             (None, 256)               131328

dropout (Dropout)           (None, 256)               0

dense_2 (Dense)             (None, 7)                 1799

=================================================================
Total params: 2,279,143
Trainable params: 2,277,735
Non-trainable params: 1,408
```

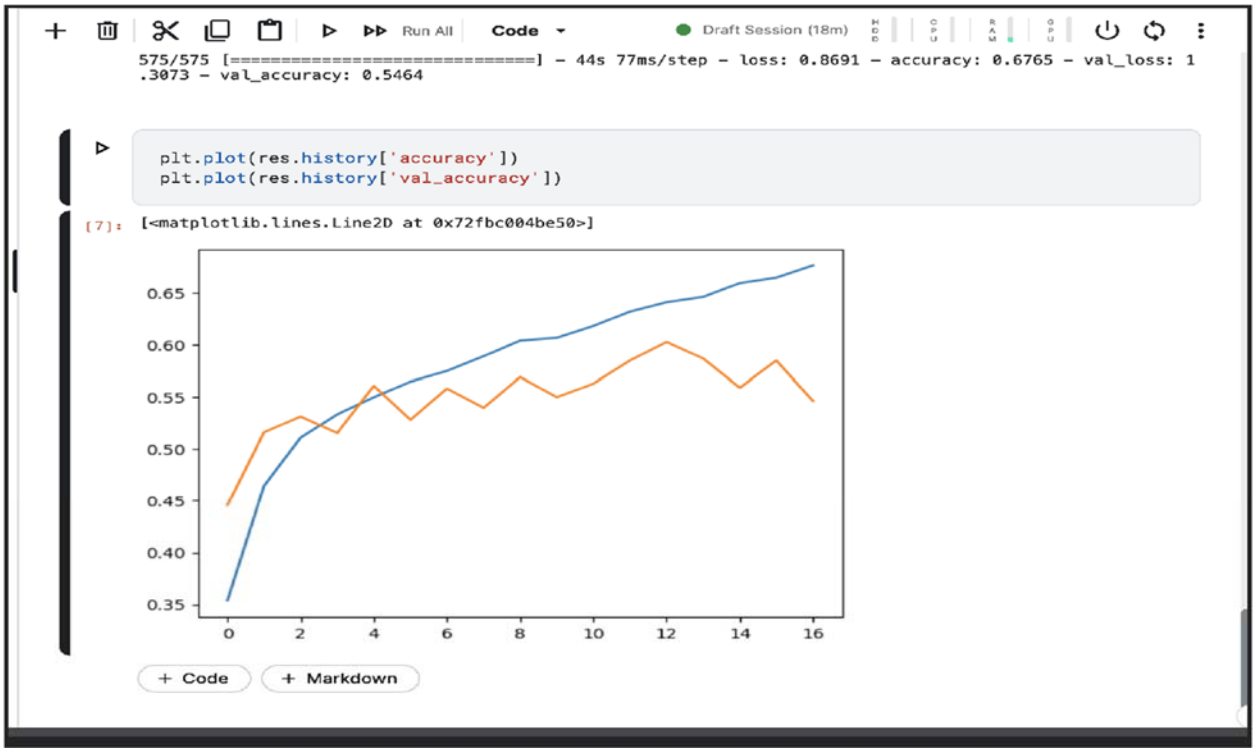
Fig. (6,7,8). Results from training version 1



```
model = Sequential()

model.add(Conv2D(32, (3, 3),padding='same', activation='relu', input_shape=(s,s, 1)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3, 3),padding='same', activation='relu'))
model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(512,(3,3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))


model.add(Flatten())
model.add(Dense(1024, activation='relu'))
model.add(Dense(512, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(7, activation='softmax')) # 7 because we have 7 classes
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 48, 48, 32)        320
```

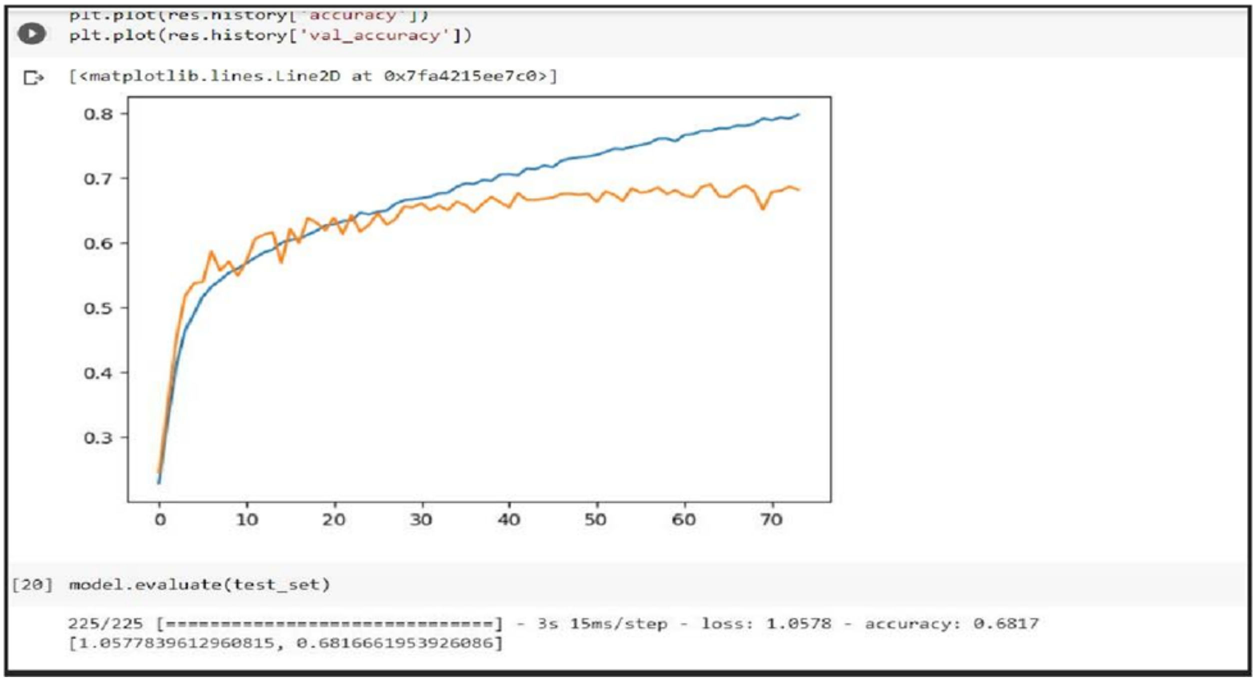
Figure (9, 10, 11). Final results



```
575/575 [==============================] - 44s 77ms/step - loss: 0.8691 - accuracy: 0.6765 - val_loss: 1
.3073 - val_accuracy: 0.5464
```

```
plt.plot(res.history['accuracy'])
plt.plot(res.history['val_accuracy'])
```

```
[7]: [<matplotlib.lines.Line2D at 0x72fbc004be50>]
```



```
plt.plot(res.history['accuracy'])
plt.plot(res.history['val_accuracy'])
```

```
[<matplotlib.lines.Line2D at 0x7fa4215ee7c0>]
```

```
[20] model.evaluate(test_set)
```

```
225/225 [==============================] - 3s 15ms/step - loss: 1.0578 - accuracy: 0.6817
[1.0577839612960815, 0.6816661953926086]
```

Conclusion

While the accuracy of facial expression recognition systems has improved significantly in recent years, achieving high levels of accuracy remains a difficult problem. A system that achieves 69% accuracy in facial expression recognition is a good first step, but there is stillroom for improvement. It is important to note that the accuracy of facial expression recognition systems can vary depending on factors such as lighting conditions, facial expressions, and individual differences.

Future Scope

Despite its limitations, a system with 69% accuracy in facial expression recognition can still be useful in applications such as psychology, human-computer interaction, and entertainment. With further research and development, we can expect to see even more accurate and effective facial expression recognition systems in the future. We aim to reach a system with 90% accuracy.

References

[1] Shan, C., Gong, S., & McOwan, P. W. (2005, September). Robust facial expression recognition using local binary patterns. In Image Processing, 2005. ICIP 2005. IEEE International Conference on (Vol. 2, pp. II-370). IEEE.

[2] Bhatt, M., Drashti, H., Rathod, M., Kirit, R., Agravat, M., & Shardul, J. (2014). A Studyof Local Binary Pattern Method for Facial Expression Detection. arXiv preprint arXiv:1405.6130.

[3] B. Fasel *et al.* *"*Automatic facial expression analysis: a survey"Pattern Recognition, (2003)

[4] M. Pantic *et al.* *"*Expert system for automatic analysis of facial expression" Image and Vision Computing, (2000).

[5] Cohen *et al.* *"*Facial expression recognition from video sequences: temporal and static modeling", Computer Vision and Image Understanding, (2003)

[6] T. Ojala *et al.* *"*A comparative study of texture measures with classification based on featured distribution", Pattern Recognition, (1996)

[7] Y. Freund *et al.* *"*A decision-theoretic generalization of on-line learning and an application to boosting", Journal of Computer and System Sciences, (1997)

[8] Xiao, X.Q.; Wei, J. Application of wavelet energy feature in facial expression recognition. In Proceedings of the 2007.

[9] Zhao, L.; Zhuang, G.; Xu, X. Facial expression recognition based on PCA and NMF. In Proceedings of the 2008