ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

EMOTION RECOGNITION USING CNN ON CK+ DATASET

PROJECT REPORT

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN
COMPUTER APPLICATIONS [TRIPLE MAIN]

Submitted By
SNEHA MOL
III BSc Computer Applications [Triple Main]
Register No: SB21CA024
Under the Guidance of
Assistant Professor
Ms.Elizabeth Paul

DEPARTMENT OF COMPUTER APPLICATIONS

2021-2024

Certificate

ST. TERESA'S COLLEGE
(AUTONOMOUS)
ESTD 1925

This is to certify that the project report entitled "EMOTION RECOGNITION USING CNN ON CK+ DATASET," is a bona-fide record of the work done by Sneha Mol (SB21CA024) during the year 2021 – 2024 and submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Applications (Triple Main) under Mahatma Gandhi University, Kottayam.

21/3/24

Head of the Department

21/03/24

Internal Examiner

22.3.24

External Examiner

Date: 21/03/24

DECLARATION

I, SNEHA MOL (Register no. SB21CA024), B.Sc. Computer Applications [Triple Main] final year student of St. Teresa's College (Autonomous), Ernakulam, hereby declare that the project submitted named "**EMOTION RECOGNITION USING CNN ON CK+ DATASET**" for the Bachelor's Degree in Computer Applications [Triple Main] is my original work. I further declare that the said work has not previously been submitted to any other university or academic body.

Date: 21/03/24

Place: Ernakulam

Sneha Mol

ACKNOWLEDGMENT

First and foremost, I thank God Almighty for his blessings. I take this opportunity to express my gratitude to all those who helped me in completing this project successfully. I wish to express my sincere gratitude to the **Provincial Superior and Manager Rev. Dr. Sr. Vinitha CSST, Director Rev. Sr Emeline CSST and Principal Dr. Alphonsa Vijaya Joseph** for providing all the facilities.

I express my sincere gratitude towards the **Head of the department Ms. Remya C J** for the support. I deeply express my sincere thanks to my guide **Ms. Elizabeth Paul** for her proper guidance and support throughout the project work. I thank all other faculty of the department for their exemplary guidance and appreciation throughout the project.

I would also like to thank all my friends and relatives for their support. I convey my hearty thanks to my family who encouraged me and provided all the opportunities for the completion of this project.

Sneha Mol

SYNOPSIS

This project investigated the potential of Convolutional Neural Networks (CNNs) for recognizing emotions from facial expressions. The model, trained on the controlled CK+ dataset, achieved good accuracy in identifying basic emotions like happiness, sadness, and anger. However, limitations emerged when tested on personal images, highlighting the need for diverse data to handle real-world variations. Future advancements could focus on creating a more robust model by expanding the training data and exploring functionalities like group emotion recognition, real-time applications, and color image processing. This project lays the groundwork for further development in emotion recognition technology, paving the way for its integration into various practical applications.

Contents

1.INTRODUCTION
1.1 About Project..2
1.2 Objectives of the Project..2
2. SYSTEM ANALYSIS
2.1 Introduction 4
2.2 Existing System...4
2.3 Proposed System 4
2.4 System Specification...5
2.5 Operating System 5
2.6 Languages or Software Packages..6
3. SYSTEM DESIGN
3.1 Introduction 8
3.2 System Design 8
3.3 Model Architecture 8
4. SYSTEM DEVELOPMENT
4.1 Data Acquisition and Preprocessing.. 10
4.2 Model Design 11
4.3 Model Training……….. 11
4.4 Model Evaluation…………... 12
4.5 Visualisation..13
5. SOURCE CODE
5.1 Model Architecture……..15
5.2 External Image…...………..16
6. RESULT
6.1 Model Evaluation..18
6.2 Model Testing...18
6.3 External Image..18
6.4 Conclusion..19
6.5 Future Scope...19
7.BIBLIOGRAPHY

1.INTRODUCTION

1.1 About Project

Emotion Recognition using CNN on the CK+ Dataset is a project about using advanced computer techniques to understand emotions from people's faces. We're using CK+ Dataset, which shows seven different expressions like anger, contempt, disgust, fear, happy, sad and surprises. By teaching computers to recognize patterns in these pictures, we hope to help in areas like making computers understand us better and improving mental health tools. This project aims to make technology more empathetic and useful for everyone.

1.2 Objectives of the project

This project aimed to explore the potential of Convolutional Neural Networks (CNNs) for recognizing emotions in facial expressions. It utilised the publicly available CK+ dataset and focused on evaluating the effectiveness of CNNs in this task.

2.SYSTEM ANALYSIS

2.1 Introduction

System analysis is a crucial process aimed at understanding, evaluating, and improving the functionality of a system. Its primary objective is to identify any existing problems, weaknesses, or inefficiencies within the system and develop strategies to address them effectively. System analysis can also be instrumental in designing and implementing new systems or integrating existing systems with new technologies or processes. By meticulously examining the components and interactions within a system, system analysis helps enhance its performance, reliability, and overall effectiveness. In today's dynamic and competitive environment, system analysis remains indispensable for organisations and businesses striving to adapt, innovate, and maintain a competitive edge.

2.2 Existing System

While there are existing systems for emotion recognition, they often face limitations. Some rely on simpler algorithms that might not achieve high accuracy, while others struggle with limited datasets or require specific hardware or software configurations. Additionally, privacy concerns can arise when dealing with personal data such as facial expressions, highlighting the need for ethical considerations in such systems. This project aimed to address these limitations by exploring a deep learning approach with readily available data, fostering further research in this domain and potentially paving the way for more robust and ethical emotion recognition solutions.

2.3 Proposed System

This project proposes a CNN-based emotion recognition system that leverages the power of deep learning for accurate and efficient emotion classification. Unlike simpler algorithms, CNNs excel at extracting intricate features from images, making them ideal for analysing the subtle nuances of facial expressions associated with different emotions.

The system utilises the CK+ dataset, a publicly available collection of labelled facial images representing various emotions like anger, happiness, and sadness. To potentially enhance model performance and address the limitations of smaller datasets, the project explores the data augmentation technique of image mirroring. This technique effectively doubles the dataset size by creating mirrored versions of existing images, providing the model with more learning opportunities.

It's crucial to highlight that this project lays the groundwork for further exploration. The proposed system requires further refinement and optimization to ensure robustness and address potential ethical concerns related to the use of personal data and privacy. Additionally, research into incorporating more diverse datasets and advanced network architectures can further enhance the system's accuracy and generalizability.

2.4 System Specification

The proposed emotion recognition system primarily relies on software components. The core element is a Convolutional Neural Network (CNN) model designed and trained specifically for this task. The CNN architecture includes 3 convolutional layers, paired with max-pooling layers, to extract features from the images. Additional layers process and classify these features, ultimately predicting the corresponding emotion.

The system operates within a cloud-based environment like Google Colab, providing access to computational resources required for training complex deep learning models. Specific hardware specifications are not crucial for this system as it primarily focuses on software implementation.

Data-related specifications are essential. The system utilises the CK+ dataset, containing images labelled with various emotions. Additionally, data augmentation techniques like image mirroring are employed to increase the effective dataset size.

2.5 Operating System

The proposed emotion recognition system primarily operates within a cloud-based environment like Google Colab. While Google Colab provides pre-configured virtual machines, their specific operating system details (e.g., Windows, Linux version) are less critical for the system itself. The system's functionality primarily relies on software libraries and frameworks like TensorFlow or Keras, which are generally compatible across different operating systems.

2.6 Languages or Software Packages

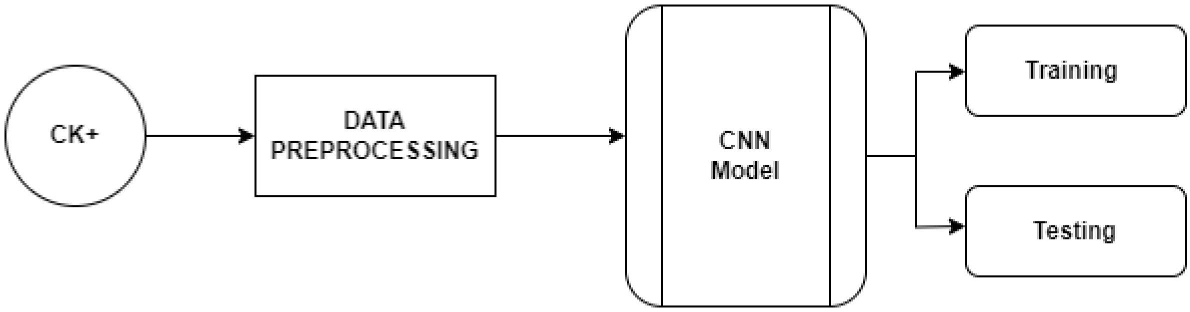
- **Python** serves as the primary programming language due to its versatility, extensive libraries, and support for deep learning frameworks.
- **Google Colab**, a cloud-based platform, is utilized for its convenience and access to powerful GPU resources, facilitating model training and experimentation.
- **OpenCV (cv2):** This computer vision library provides essential functionalities for image processing tasks crucial for your CNN model. It allows us to read and manipulate images, perform necessary transformations (resizing, normalisation), and potentially aid in image augmentation techniques like mirroring.
- **NumPy:** This fundamental library offers efficient tools for working with numerical data, including arrays and matrices. It plays a significant role in manipulating and preparing your image data for the CNN model
- **Scikit-learn (sklearn):** This machine learning library provides helpful functions for data splitting, a crucial step in training and evaluating the CNN model. It was used to split the dataset into training, testing, and validation sets.
- **TensorFlow:** This deep learning library is the foundation for building and training the CNN model. It offers a comprehensive toolbox for defining and training neural networks, performing computations, and managing the training process.
- **Matplotlib:** This library helps visualise data, allowing us to explore the image data and potentially visualise the performance of the CNN model.

3. SYSTEM DESIGN

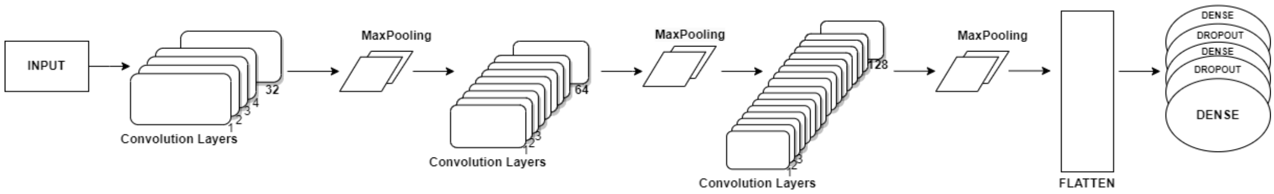
3.1 Introduction

The proposed emotion recognition system utilizes a Convolutional Neural Network (CNN) architecture to analyze facial expressions and predict emotions. This section delves into the system design, outlining the architecture, key components, and functionalities that enable the system to achieve its goals.

3.2 System design



3.3 Model Architecture



4. SYSTEM DEVELOPMENT

4.1 Data Acquisition and Preprocessing:

- Data collection: Collected CK+ dataset that has images labelled with corresponding emotions from kaggle.
- Data Preparation : We first prepared the images for the model. We made them all the same size (like 48x48 pixels) and turned them into grayscale. We also adjusted the brightness of the images (from 0-255 to 0-1) to make them more consistent. We could even have made more copies of the images by slightly mirroring them to help the model learn better.
- Data splitting:

  Ratio definition: The dataset was divided into three portions:

  - **Training set:** This set accounts for **80%** of the data and is used to train the model. During training, the model learns to identify patterns and relationships within the data that differentiate between the various emotions.
  - **Validation set:** This set comprises **10%** of the data and is used to monitor the model's performance during the training process. It helps prevent overfitting, where the model performs well on the training data but poorly on unseen data.
  - **Testing set:** This final set also accounts for **10%** of the data and is used to evaluate the model's final performance on unseen data after the training process is complete. This allows for an unbiased assessment of the model's generalizability.

  Splitting process: The code utilizes the `train_test_split` function from the scikit-learn library to randomly divide the dataset into the training and testing sets. Subsequently, a further split is performed on the remaining data to create the validation set, ensuring a representative distribution of emotions across all three sets.

4.2 Model Design and Implementation:

- Model selection: We chose a specific type of deep learning model called a Convolutional Neural Network (CNN) for this project. CNNs are particularly well-suited for tasks involving images because they can automatically learn to extract important features directly from the data. This eliminates the need for manual feature engineering, which can be a time-consuming and complex process. In essence, the CNN "learns to see" the patterns and details within the images that are most relevant to distinguishing between different emotions.
- Model development using TensorFlow and Keras:
  - **Convolutional layers:** These layers, with varying numbers of filters (e.g., 32, 64), are responsible for extracting features from the images at different levels of detail.
  - **Pooling layers:** These layers are used to reduce the dimensionality of the data, making the model more efficient while preserving relevant information.
  - **Dense layers:** These layers perform the final classification task, where the model learns to map extracted features to the corresponding emotion categories. The final output layer has 7 units, each representing one of the seven emotions the model aims to predict.

To fine-tune the model's learning process, we configured various parameters like the number of filters in each convolutional layer, activation functions (e.g., ReLU), and the number of units in the dense layers. Additionally, an optimizer (e.g., Adam) and a loss function (e.g., categorical crossentropy) were chosen and used during the training process.

4.3 Model Training:

1. Feeding Data in Batches: Instead of overwhelming the model with all the data at once, I presented it with smaller groups of images (batches) and their labels

at each training step. This allows the model to process the information more efficiently and update its internal parameters more effectively.

2. Learning Through Backpropagation: As the model processed each batch, it compared its predictions for the emotions with the actual labels provided. If the predictions were incorrect, an error signal was generated. This signal was then used in a technique called backpropagation to adjust the model's internal parameters (weights and biases) in a way that reduces the error and improves the accuracy of its predictions in the next batch.
3. Minimizing the Loss: The loss function measures the difference between the model's predictions and the actual labels. During training, I aimed to minimize this loss function by adjusting the weights and biases through backpropagation. As the loss decreased over successive training epochs (iterations), it indicated the model was gradually learning to map the features extracted from the images to the correct emotions.

In my training process, I specified 20 epochs, meaning the entire training set was passed through the model 20 times. Additionally, I used a batch size of 32, meaning 32 images and their labels were processed at a time. These are hyperparameters, and their optimal values can be further fine-tuned through experimentation to potentially improve the model's performance.

4.4 Model Evaluation:

- Feeding the Test Set.
- Performance Metrics: The model then generated predictions for the emotions in the test set images. I employed two key metrics to evaluate the model's performance:
  - Accuracy: This metric measures the percentage of correctly predicted emotions compared to the actual labels in the test set. In your specific case, the model achieved an accuracy of 97.97%, indicating it accurately predicted emotions for nearly 98% of the images in the unseen test set.

- Loss: The loss metric reflects the difference between the model's predictions and the actual labels. A lower loss value indicates better alignment between the model's predictions and the true emotions. In your case, the model achieved a test loss of 0.1137, suggesting a good level of agreement between its predictions and the actual emotions.

4.5 Visualisation :

I utilized Matplotlib to perform a **qualitative assessment** of the model's performance by predicting the emotion on a single image selected from the test set.

- **Random Selection:** I chose a random image from the test set using numpy.randint.
- **True Label Extraction:** I retrieved the true emotion label associated with the selected image from the y_test array.
- **Image Reshaping:** The image was reshaped to match the input shape the model expects for prediction.
- **Model Prediction:** I fed the reshaped image into the trained model, and it generated its prediction for the emotion in the image.
- **Predicted Label Extraction:** I identified the specific emotion predicted by the model using the argmax function, which finds the index of the highest value in the prediction array.
- **Emotion Label Definition:** I created a list of emotion labels corresponding to the model's output, ensuring clarity in the visualisation.
- **Visualization:** Using Matplotlib, I displayed the chosen image in grayscale format and overlaid the true label and the model's predicted label. This allowed for a visual comparison of the model's performance on a specific example.

5. SOURCE CODE

5.1 Model Architecture

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

input_shape = (48, 48, 1)

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=input_shape),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(64, activation='relu'),
    Dropout(0.5),
    Dense(7, activation='softmax')
])

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])


model.summary()
```

5.2 External Image

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files

# Function to preprocess the uploaded image
def preprocess_image(image_path, target_size=(48, 48)):
    img = cv2.imread(image_path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  # Convert to grayscale
    img = cv2.resize(img, target_size)  # Resize image to target size
    img = img.astype('float32') / 255.0  # Normalize pixel values
    img = np.expand_dims(img, axis=0)  # Add batch dimension
    img = np.expand_dims(img, axis=-1)  # Add channel dimension
    return img

# Upload an image using Colab's file upload feature
uploaded = files.upload()

# Assuming only one image is uploaded, get its path
uploaded_image_path = list(uploaded.keys())[0]

# Preprocess the uploaded image
processed_image = preprocess_image(uploaded_image_path)

# Use the trained model for prediction
emotion_labels = ['anger', 'contempt', 'disgust', 'fear', 'happy', 'sadness', 'surprise']
prediction = model.predict(processed_image)
predicted_emotion = emotion_labels[np.argmax(prediction)]

# Load and display the uploaded image
uploaded_image = cv2.imread(uploaded_image_path)
uploaded_image_rgb = cv2.cvtColor(uploaded_image, cv2.COLOR_BGR2RGB)
plt.imshow(uploaded_image_rgb)
plt.axis('off')
plt.title('Predicted Emotion: {predicted_emotion}')
plt.show()
```

6. RESULT

6.1 Model Evaluation

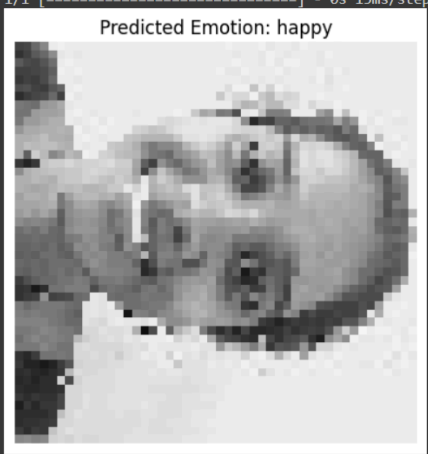
```
7/7 [==============================] - 0s 28ms/step - loss: 0.1109 - accuracy: 0.9746
Test Loss: 0.11092516034841537
Test Accuracy: 0.9746192693710327
```

6.2 Model Testing

```
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 13ms/step
1/1 [==============================] - 0s 14ms/step
1/1 [==============================] - 0s 14ms/step
```

6.3 External image

```
Choose Files gowri smile.jpg
• gowri smile.jpg(image/jpeg) - 1844 bytes, last modified: 2/15/2024 - 100% done
Saving gowri smile.jpg to gowri smile.jpg
1/1 [==============================] - 0s 15ms/step
```



6.4 Conclusion

This project successfully developed a foundational model for emotion recognition using grayscale facial images from the CK+ dataset. The model achieved an impressive accuracy of 97% on the unseen test set, demonstrating its capability to learn and generalize to new data. However, when tested on personal images with potentially less controlled expressions, the model faced challenges, highlighting the impact of data distribution on its performance.

6.5 Future Scope

Advancements such as group emotion recognition, integrating object detection and emotion analysis techniques, as well as optimising the model for real-time applications, can expand its functionality for various settings. Additionally, extending the capability to process colour images can enrich the feature extraction process and potentially improve recognition accuracy. Through investigating and implementing these advancements, the project lays the groundwork for further development in emotion recognition technology, paving the way for its integration into various real-world applications.

7. BIBLIOGRAPHY

7.1 References

1. Cîrneanu, Andrada-Livia, Dan Popescu, and Dragoș Iordache. "New trends in emotion recognition using image analysis by neural networks, a systematic review." *Sensors* 23.16 (2023): 7092.
2. Abdullah, Sharmeen M. Saleem, and Adnan Mohsin Abdulazeez. "Facial expression recognition based on deep learning convolution neural network: A review." *Journal of Soft Computing and Data Mining* 2.1 (2021): 53-65.
3. Meena, Gaurav, et al. "Identifying emotions from facial expressions using a deep convolutional neural network-based approach." *Multimedia Tools and Applications* (2023): 1-22.
4. Madhav, Govind, and Faisal Rais. "EMOTİON RECOGNİTİON USİNG EXPLİCİT FEATURE EXTRACTİON METHOD."
5. **Emotion Detection Model using CNN — a complete guide:** https://medium.com/@skillcate/emotion-detection-model-using-cnn-a-complete-guide-831db1421fae
6. **Facial Emotion Recognition using Deep Learning A thesis submitted in partial fulfillment:** https://scholarworks.calstate.edu/concern/theses/dz010x52t