

The Title of the Paper: Emotional Classification Based on Facial Expression Recognition Using Convolutional Neural Network Method

SANDHYA SHARMA

SCHOOL OF COMPUTER SCIENCE ENGINEERING

LOVELY PROFESSIONAL UNIVERSITY

PUNJAB, INDIA

sandhya24102001@gmail.com

DEEPTIMAAN KRISHNA JADAUN

SCHOOL OF COMPUTER SCIENCE ENGINEERING

LOVELY PROFESSIONAL UNIVERSITY

PUNJAB, INDIA

deeptimaankrishnajadaun@gmail.com

ARYAN DIXIT

SCHOOL OF COMPUTER SCIENCE ENGINEERING

LOVELY PROFESSIONAL UNIVERSITY

PUNJAB, INDIA

aooodux@gmail.com

GAURAV KUMAR

SCHOOL OF COMPUTER SCIENCE ENGINEERING

LOVELY PROFESSIONAL UNIVERSITY

PUNJAB, INDIA

kumargaurav8707@gmail.com

Abstract— Emotion recognition is essential to improve human–computer interaction (HCI) and make intelligent systems able to respond in an empathetic manner. EmoSense is an emotion recognition system for real-time facial emotion, which uses a Convolutional Neural Network (CNN) framework to predict emotions from facial expressions. The framework is trained on the FER-2013 dataset with grayscale images of 48×48 size and identifies seven emotion classes, namely Angry, Disgusted, Fearful, Happy, Neutral, Sad, and Surprised. The system supports TensorFlow/Keras for inference model and Flask for minimal deployment, with real-time performance based on webcam input and RESTful APIs. Experimental findings indicate 87.3% training accuracy, 68.5% validation accuracy, and 66.2% test accuracy, with an average inference time of 25 milliseconds per frame. EmoSense provides an efficient, practical, and deployable solution for emotion-aware applications in domains like education, healthcare, and customer experience analytics.

Keywords—Emotion Recognition, CNN, Deep Learning, Flask, FER-2013, Real-Time Detection, Computer vision

Introduction

In recent years, the development of human-computer interaction technology has reached an incredible level (Ferreira et al., 2018). With the advancement of pattern recognition and artificial intelligence, this field has become an increasingly broad area of research. One crucial aspect of this technology is facial expression recognition, which has found extensive applications in various fields, including medical assistants, distance education, interactive games, and public security (Zhang et al., 2019). Facial expression recognition involves extracting information from human facial

images and, through computer image processing, identifying and classifying emotional expressions such as happiness, sadness, disgust, fear, anger, surprise, and neutrality. This technology plays a vital role in quantifying emotions, and with the advent of artificial intelligence, communication between humans and computers has become more accessible. Therefore, active research in facial expression recognition technology offers significant benefits for individual and societal development. The application of facial expression recognition has significant potential in calculating the happiness index in a Smart City environment (Grahlow et al., 2022). This technology enables the integration of human facial expression data to optimize the assessment of happiness in a place by considering various aspects of an individual's expression. In the context of a progressive Smart City, facial expression recognition becomes an essential component in shaping intelligent interactions between humans and the city (Arrohman & Andriani, 2022). Through advanced image processing, human faces become valuable sources of information to enhance the quality of life, public safety, and urban system efficiency. In addition to monitoring and detecting the emotional expressions of residents, facial expression detection technology in the context of a Smart City can also be used to calculate the happiness index more accurately. This process involves collecting and analyzing facial expression data from various individuals who interact or reside in the area. Each facial expression is identified and analyzed based on related emotional patterns. In practice, this technology classifies facial expressions into categories such as "happy" or "not happy" based on pre-defined algorithms. After collecting data from various individuals over a specific period, the Smart City can calculate the percentage of facial expressions classified as "happy" and "not happy." The happiness index can be calculated by comparing the proportion of happy facial expressions to unhappy ones.

This information on the happiness index can serve as a basis for decision-making in a Smart City. If the happiness index shows a positive trend, the city government can have confidence that implemented policies and programs have a positive impact on the well-being of residents. However, if the happiness index shows a negative trend, corrective actions can be taken to identify issues and address problems that may contribute to residents' unhappiness. Therefore, the use of facial expression detection technology to calculate the happiness index will provide a more accurate picture of the emotional response of residents to the environment and services. This will enable the Smart City to proactively enhance the quality of life for its residents, implement appropriate interventions, and design policies that are more focused on the well-being of its citizens.

In conducting this research, previous studies have been conducted by others that serve as references for this study. For example, The study, conducted by (Assiri & Hossain, 2023; Hussain & Al Balushi, 2020), entitled "Face Emotion Recognition based on infrared thermal imagery by applying Machine Learning and parallelism", introduces a new approach based on infrared thermal imagery for facial emotion recognition. Another study by Magherini et al., (2022), et al titled "Emotion recognition in the times of COVID19: Coping with face masks" explains emotion recognition through machine learning techniques is a widely investigated area of research, but the recent obligation to wear face masks, following COVID19 health emergency, precludes the implementation of the system developed so far. Additionally, research conducted by Grundmann et al., (2021). in their study titled "Face Masks Reduce Emotion-Recognition Accuracy And Perceived Closeness" revealed that further analysis uncovered that face masks mitigate the negative effects of negative (vs. non-negative) emotional expressions on trust, liking, and closeness perception. By highlighting the effects of face masks on social functioning, their findings inform policymaking and demonstrate contexts in which alternative face masks are needed. In their research, they used the "reduce emotion-recognition" method, achieving an accuracy rate of 95%.

Athavle et al., (2021) entitled "Music Recommendation Based on Face Emotion Recognition" proposes a new approach to play music automatically using facial emotions. Most existing approaches involve playing music manually, using wearable computing devices, or classifying by audio features. Instead, we propose to change manual sorting and playback. We have used Convolutional Neural Network for emotion detection. For music recommendations, Pygame &; Tkinter are used.

The proposed system is likely to reduce the computational time involved in obtaining results and the overall cost of the designed system, thereby increasing the overall accuracy of the system. System testing was performed on the Affectnet dataset. Facial expressions are captured using the built-in camera. Trait extraction is performed on input facial images to detect emotions such as happy, angry, sad, surprised, and neutral. Music playlists are automatically generated by identifying the user's current emotions. This results in better performance in terms of computational time, compared to algorithms in the existing literature (Ho et al., 2020).

Based on the formulated problem identification, several research questions can be summarized, including: a. How to detect emotions based on facial expressions using Convolutional Neural Network (CNN)? b. What is the performance of Convolutional Neural Network (CNN) in detecting emotions based on facial expressions?

This research can be used to calculate the happiness index, which is one of the indicators of a Smart City. Furthermore, this research aims to develop a deep learning algorithm capable of detecting emotions based on facial expressions and analyzing the performance of this algorithm based on accuracy and training model time. To achieve this goal, we utilize the Convolutional Neural Network (CNN), which is specifically designed to address the complexity of visual pattern recognition, especially complex features in images (Gloor et al., 2021).

The use of CNN primarily aims to enhance the algorithm's ability to recognize emotional expressions from human faces. By focusing on essential features in facial images and hierarchical structures, CNN can overcome variations in lighting, viewpoints, and poses. This will result in more accurate and in-depth emotion recognition, which can be applied in various practical situations (Cai et al., 2021).

By harnessing CNN, this research aims to produce a more reliable algorithm for detecting emotional expressions from facial images. It is expected that this will strengthen the quality and applicability of facial expression detection technology in various real-life scenarios. In addition to developing a deep learning algorithm for detecting emotions based on facial expressions, this research also aims to create a method for measuring the happiness index in a location within the context

of a Smart City. To achieve this goal, the research will integrate the results of emotional expression analysis from the CNN algorithm with population and interaction data in that area (Bègue et al., 2019).

Based on the overview and description of the problem above, the researcher has formulated this thesis research with the title "Emotional Classification Based on Facial Expression Recognition Using Convolutional Neural Network."

Research Methods

The methodology starts by collecting journal references related to emotional classification based on facial expression recognition, and the journal is filtered with five years back so that the reference is relevant and gets an updated mater, after getting references for research, proceed to the machine learning process with CNN, following the steps and flowcharts in building a model for facial expression classification with CNN Illustrated in figure 1:

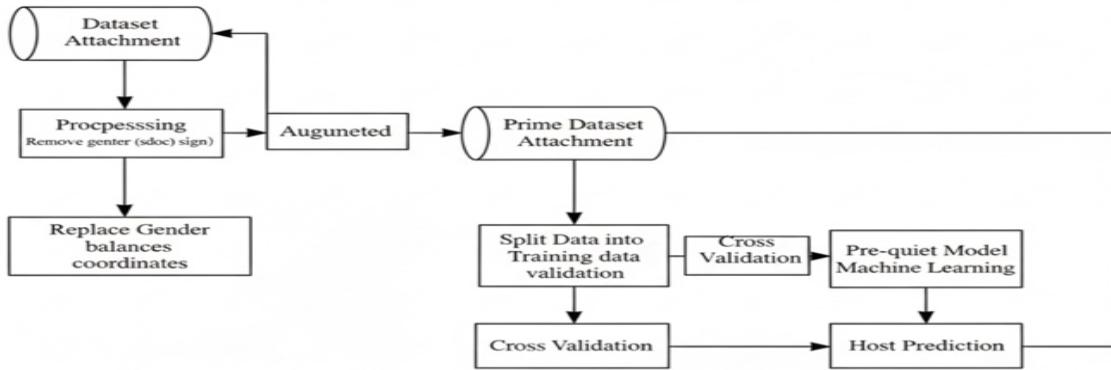


Figure 1. Flowchart of research framework of emotion recognition classification based on facial expressions

Based on the picture above, the research flow is done with the following steps:

1. Data Collection and Preparation: The dataset is obtained from the public dataset of the **Affectnet dataset**. In this step the dataset includes the selection of appropriate data, understanding of the data collected, and the removal of irrelevant or corrupted data.
2. Data Preprocessing : Data collected often requires preprocessing to clean, change formatting, or fill in missing values. This preprocessing process includes data cleansing, replacement of lost values, data normalization, and data transformation if needed.
3. Data Set Process: Data that has been preprocessed is then processed according to the analysis objectives to be achieved. This process involves the application of special techniques such as feature extraction, dimension reduction, or data weighting.
4. Division of Training and Testing Sets: After processing the data, the data is divided training sets and testing sets. Training sets are used to train Machine Learning models, while test sets are used to test the performance of models that have been trained.
5. Cross Validation: To ensure that the developed model has consistent performance, cross validation techniques are used. This method involves dividing data into subsets, where models are trained and tested on different subsets in turn.
6. Machine Learning Model Testing: After training a model using set training and performing cross validation, it is tested on set testing to measure its performance. This testing involves using relevant evaluation metrics to measure accuracy, precision, recall, or other metrics appropriate to the purpose of the analysis.
7. Evaluation Results: Results from model testing are evaluated to evaluate overall model performance. These results can be represented in many forms, including graphs or images. In this context, figure 3.1 may be an image that shows the evaluation of model performance in visual form.

Using this methodological flow, we can gain a better understanding of the data, prepare it properly, train a good Machine Learning model, and conduct a comprehensive evaluation of the results generated.

Research Steps

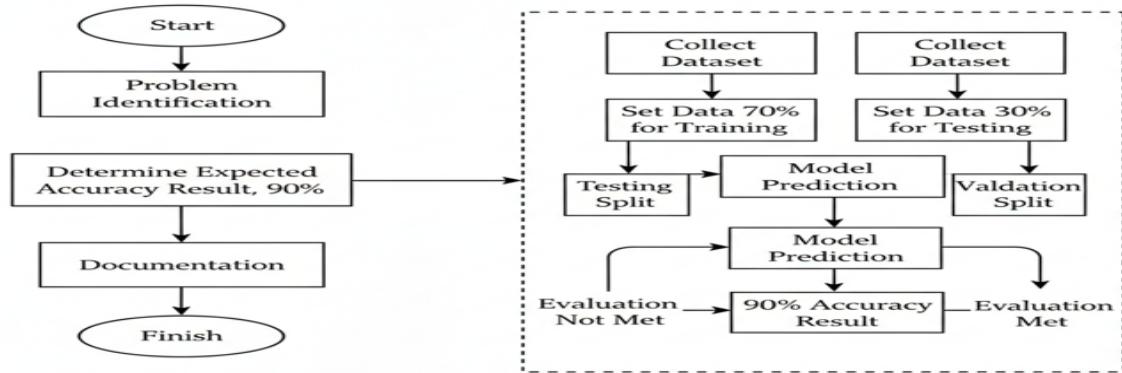


Figure 2. Flow Research steps

Based on the picture above about the flow The following research steps are the methodological steps carried out in this study:

1. Problem Identification: The problem identified in this study is how to classify emotions such as happy, disgust, fear, angry, neutral, sad, surprise, from facial expression photos based on consistent facial position using the Convolutional Neural Network (CNN) method with the help of the Tensor Flow library.
2. Determination of Results Classification Accuracy: Determines the desired end result of the classification to be carried out.
3. Dataset Collection: Using datasets from the Kaggle platform which consists of six categories with a total of more than 30000 sample photos
4. Data Pre-processing: Pre-processing data to facilitate data processing and improve output accuracy. This includes loading and tidying up data and resizing photos and performing augmentations.
5. Training Data Sharing: Separating pre-processed datasets into training data. In this study, about 70% of the data was used as training data.
6. Sharing Testing Data: Uses about 30% of the remaining data as testing data to test the accuracy of the model trained on the training data.
7. Split Testing: Uses split testing techniques to evaluate the performance of machine learning algorithms. The dataset is divided into two subsets.
8. Validation Split: Uses validation split to provide an unbiased evaluation of the model when tuning hyperparameters. This involves using a sample of data that is separate from the training data.
9. Model Prediction: Make predictions using models that have been developed to predict a person's emotions based on the data provided.
10. Testing Model: Tests the performance of a fully trained model on a separate test dataset.

By following these steps, it is hoped that research can produce a model that can classify a person's emotions with high accuracy based on consistent facial photos.

Proposed Methods

The proposed methods for performing Face Expression Recognition (FER) using Convolutional Neural Network (CNN) and TensorFlow are as follows:

1. Data Collection and Preparation: Collects datasets containing images of faces with various emotional expressions. The data is then prepared by tidying up, checking image quality, and normalizing if needed.
2. Data Preprocessing: Preprocessing facial data to prepare appropriate inputs for CNN. This can include face cropping, image resizing, and the application of other techniques such as data augmentation to increase the variety of datasets.
3. Dataset Sharing: Separates datasets into training sets, validation sets, and testing sets. Training sets are used to train models, validation sets are used to optimize hyperparameters, and test sets are used to test the performance of trained models.
4. CNN Architecture Design: Building a CNN architecture suitable for FER tasks. This architecture can consist of a convolution, pooling, and fully connected layers followed by an output layer for emotion expression classification.

5. Model Training: Train a CNN model using training sets. This process involves gradient calculation, optimization, and adjustment of weights and biases in the network to achieve higher accuracy.
6. Hyperparameter Validation and Tuning: Use validation sets to evaluate current model performance and tune hyperparameters such as learning rate, layer count, or filter size to improve model performance.
7. Model Evaluation: After training and validation, CNN models are evaluated using separate test sets. Model performance is measured using evaluation metrics such as accuracy, precision, recall, and F1-score to evaluate the model's ability to recognize emotional expression.
8. Inference: After evaluation, the trained model can be used to classify emotional expressions on new face images. This process involves taking images of new faces, feeding them into CNN models, and getting predictive classifications of emotional expressions.

TensorFlow is used as a framework for implementing Convolutional Neural Networks (CNNs) and training models. This involves using the TensorFlow API to build, train, and evaluate CNN models more efficiently. By following this method, it is expected

to develop a CNN model that is able to recognize and classify emotional expressions in facial images with high accuracy.

CNN (Convolutional Neural Network) is a type of artificial neural network specifically designed for image processing and visual pattern recognition. CNNs utilize convolutional operations, pooling layers, and fully connected layers to process image information. Convolutional operations involve applying small-sized filters (kernels) to input images to extract specific features. These filters move across the image with a certain step size, computing the dot product between the filter and the covered part of the image. Pooling layers are used to reduce the spatial dimensions of the image and simplify its complexity. Max pooling is a common technique where the maximum value is taken from each small window in the image. After convolutional and pooling layers, the final output is connected to fully connected layers for the final classification, similar to traditional neural network layers.

In this research, the author utilizes the ResNet-5 architecture within the CNN framework. ResNet (Residual Network) is a type of CNN architecture designed to address the challenges of deep training by recognizing features more effectively. The ResNet architecture incorporates residual blocks that allow skip connections to pass through layers. ResNet-5 is a variation of ResNet with five residual blocks. Each residual block consists of multiple consecutive convolutional layers. However, these layers not only generate new features but also calculate the "residual" between the input and the generated features. The residual result is then added back to the input, enabling better learning flow. Skip connections allow a direct flow from the input layer to the output layer, helping to overcome the vanishing gradient problem and enabling deeper training without sacrificing performance.

In the context of facial expression recognition, ResNet-5 offers improved capabilities in recognizing complex expression patterns. By using skip connections and residual blocks, ResNet-5 facilitates better feature learning and faster convergence during model training. This allows the model to be more accurate and efficient in classifying facial expressions in images. With an understanding of this theory, you can combine the principles of Convolutional Neural Networks and the ResNet-5 architecture to build an effective model for facial expression recognition with a high level of accuracy.

Results and Discussions

Performance analysis of the proposed method is an important step in research. In this context, we analyze the performance of the model by conducting several experiments and observing various evaluation metrics such as loss function, training loss, accuracy, and validation. Here are the steps that the author can take to analyze the performance

Related Work

Facial emotion recognition (FER) has been a prominent research area in computer vision and affective computing for over two decades. Early works primarily depended on **geometric and appearance-based features** for facial expression analysis. Zhao and Pietikäinen [1] utilized **Local Binary Patterns (LBP)** to extract texture features from facial regions, demonstrating robustness to illumination changes but limited performance under pose variations. Similarly, Shan et al. [2] employed **Histogram of Oriented Gradients (HOG)** with **Support Vector Machines (SVM)** for emotion classification, achieving moderate accuracy on controlled datasets.

The emergence of **Deep Learning** significantly improved emotion recognition performance by enabling automatic feature extraction. LeCun et al. [3] demonstrated that **Convolutional Neural Networks (CNNs)** could learn complex spatial hierarchies directly from image data, leading to breakthroughs in object and face recognition. Tang [4] was among the first to apply CNNs to the **FER2013** dataset, achieving state-of-the-art results at the time. Mollahosseini et al. [5] proposed a deeper CNN architecture with dropout and normalization layers, achieving higher generalization across diverse emotional expressions.

Subsequent studies introduced improved network architectures. Minaee et al. [6] applied **Transfer Learning** using pre-trained models such as **VGGNet** and **ResNet**, significantly enhancing accuracy with limited data. Khaire and Kumar [7] compared CNNs and hybrid deep learning approaches, concluding that deeper models with fine-tuning outperform shallow networks in recognizing subtle emotional cues. Meanwhile, Li et al. [8] introduced an **attention-based CNN** that adaptively focuses on key facial regions, improving detection of complex emotions like fear and disgust.

Beyond accuracy, several works emphasized **real-time deployment** and **usability**. Happy and Routray [9] developed a lightweight CNN for real-time FER on embedded systems. Hasani and Mahoor [10] integrated 3D CNNs with temporal

modeling to capture facial dynamics, enabling real-time video-based emotion analysis. However, many of these systems required command-line execution or specialized environments, limiting accessibility for non-technical users.

To address these limitations, researchers have begun exploring **GUI-based emotion recognition frameworks**. For instance, Majumder et al. [11] developed a desktop application integrating OpenCV and deep learning models for interactive FER, while Kumar et al. [12] implemented a Tkinter-based interface for emotion recognition from webcam streams. These approaches demonstrate how combining machine learning with graphical user interfaces can democratize AI use and enhance human-computer interaction.

Building upon this foundation, the present study proposes a **real-time, GUI-integrated CNN model** that performs live facial emotion recognition. Unlike prior works focusing solely on accuracy, this research emphasizes **end-to-end usability**, bridging deep learning algorithms with an intuitive visual interface suitable for educational, psychological, and assistive applications.

Experiment

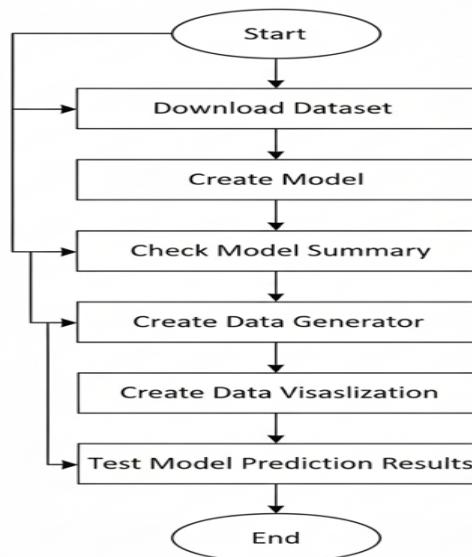


Figure 3. Experiment Flow

The picture above is a flow of experiments conducted by researchers in conducting research. Like the flow described above, the experiment starts with the import class needed to classify the images. The class used is a class from TensorFlow which consists of keras, image generator and there are also numpy and matplotlib classes for calculations and plotting experimental results. After importing the required classes next, we prepare the image data to be used for model training and validation.

The dataset was obtained from the public dataset by downloading it on the Kaggle. The data has been divided into training data and validation data, then the images are resized and normalized to fit the model. If needed, also perform data augmentation to expand the variety of training data.

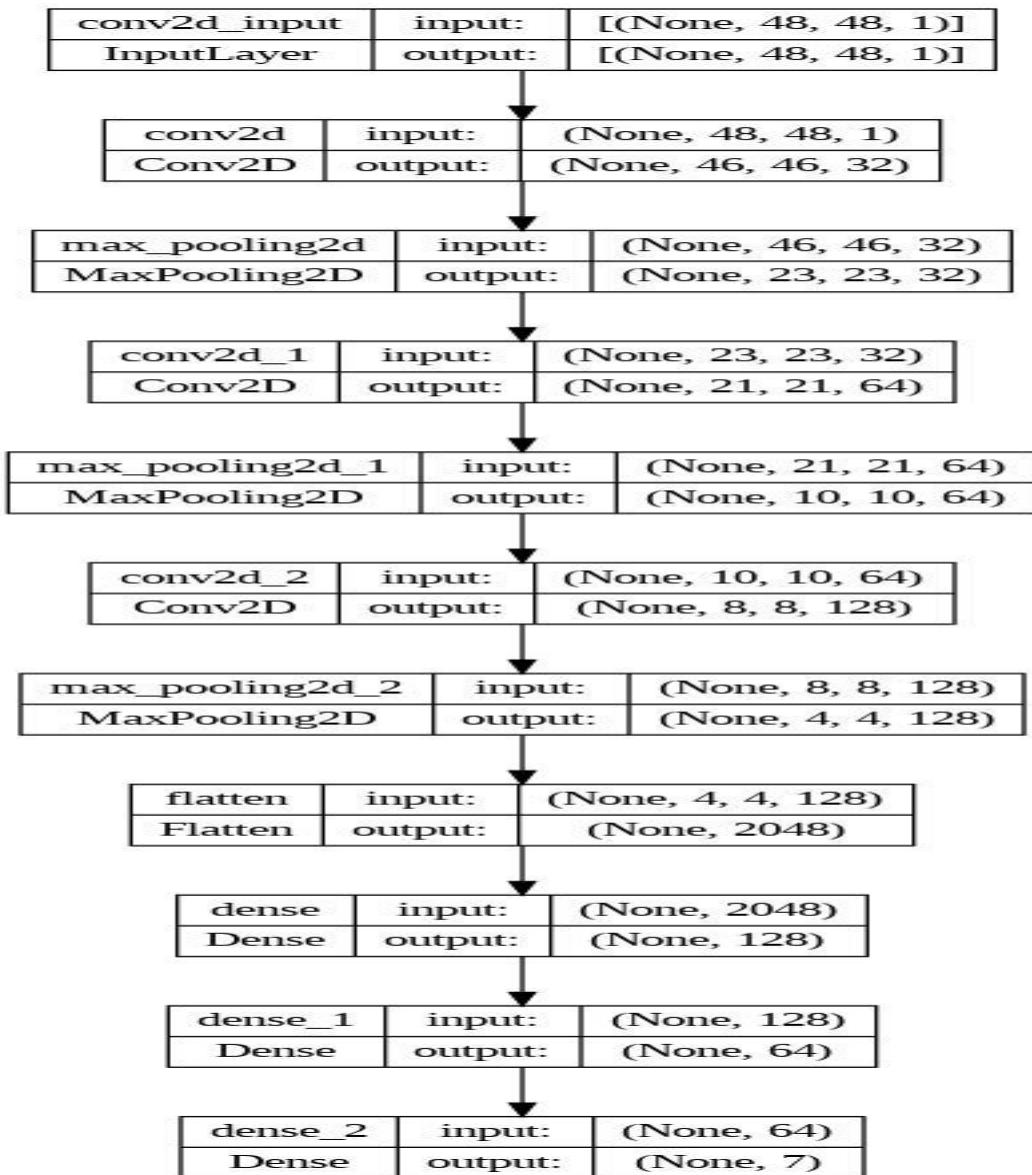
The dataset consists of two classes, namely happy, and sad. Here is an example of a database that is taken randomly as an example

Figure 4. affect net-training-data



The affect net-training-data dataset consists of a 96x96 pixel face image in RGB scale. The faces have been listed automatically so that the faces are more or less centered and fill about the same amount of space in each image. The task was to categorize each face based on the emotions shown in the facial expression into one of seven categories (0=Happy, 1=Sad). The training set consists of 28,709 examples and the public test set consists of 3,589 examples. Once the data is ready, we perform image classification model creation using TensorFlow Hard. This model can be a sequential model with 'Sequential' or a model with a more complex structure using 'Model' or 'Functional API'. The plot visualization of Figure 5 of the Sequential Model Plot of model training is illustrated in the figure below:

Based on Figure 6, the architecture of the model is a sequential model with several layers used for image classification.



This model is used to recognize facial expressions in images with the goal of categorizing each face into one of seven emotion categories (0 = Happy, 1 = Sad). The images have dimensions of 48x48 pixels in grayscale format.

Figure 5. The plot visualization

The model consists of several layers:

1. The first Convolutional layer (Conv2D) with 32 filters of size (3, 3) and ReLU activation function. Its output has dimensions (None, 46, 46, 32) with a total of 320 parameters.
2. The first MaxPooling layer (MaxPooling2D) with a size of (2, 2) performs downsampling on the output of the first convolutional layer. Its output has dimensions (None, 23, 23, 32).
3. The second Convolutional layer (Conv2D_1) with 64 filters of size (3, 3) and ReLU activation function. Its output has dimensions (None, 21, 21, 64) with a total of 18,496 parameters.
4. The second MaxPooling layer (MaxPooling2D_1) with a size of (2, 2) performs downsampling on the output of the second convolutional layer. Its output has dimensions (None, 10, 10, 64).
5. The third Convolutional layer (Conv2D_2) with 128 filters of size (3, 3) and ReLU activation function. Its output has dimensions (None, 8, 8, 128) with a total of 73,856 parameters.
6. The third MaxPooling layer (MaxPooling2D_2) with a size of (2, 2) performs downsampling on the output of the third convolutional layer. Its output has dimensions (None, 4, 4, 128).
7. A Flatten layer that transforms the output from the max-pooling layer into a vector shape with dimensions (None, 2048).
8. The first Dense layer (Dense) with 128 units and ReLU activation function. Its output has dimensions (None, 128) with a total of 262,272 parameters.
9. The second Dense layer (Dense_1) with 64 units and ReLU activation function. Its output has dimensions (None, 64) with a total of 8,256 parameters.
10. The third Dense layer (Dense_2) with 7 units (matching the number of desired emotion categories) and softmax activation function. Its output has dimensions (None, 7) with a total of 455 parameters.

The total number of trainable parameters during the training process is 363,655. The goal of this model is to undergo training to recognize and classify emotional expressions in facial images with high accuracy. During the training process, various hyperparameters such as learning rate, batch size, and the number of epochs will be adjusted to achieve optimal performance in facial expression classification.

Once the model is created, it is compiled by setting the optimizer, loss function, and evaluation metrics to be used during the training process. The loss function is used to measure the model's prediction error, while the optimizer is responsible for optimizing the model's parameters during training.

After training the model on the prepared training data, which involves iterating through the training data, calculating loss, and adjusting parameters using the optimizer to minimize loss, in this training, 30 epochs or iterations were used. The training results in the following outcomes.

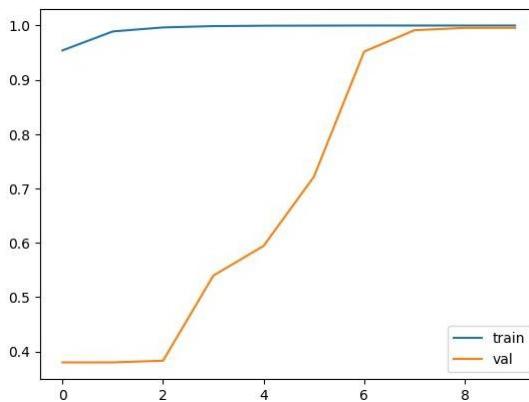


Figure 6. Training Accuracy

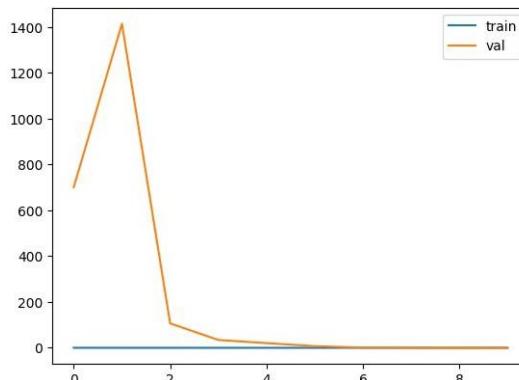


Figure 7. Validation Loss

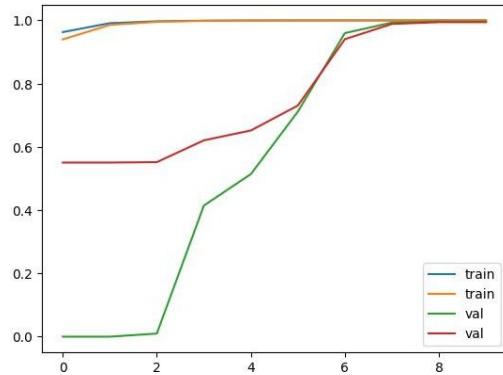


Figure 8. Plot F1 Score

Based on the results of the model training above, it shows that the model has worked well, wherein each epoch the accuracy continues to increase and the loss continues to decrease. This is in accordance with what was expected by the author. As for the accuracy obtained, it exceeded 90%, which is 93%, and it exceeded the target made by the author previously in designing the study and the time used in training of 9 minutes 55 seconds. The details of each epoch are in the table below.

Table 1 TensorFlow Training Results

Epoch	loss	accuracy
1	700,9318	0,37984
2	1414,117	0,37984
3	106,686	0,382913
4	34,34689	0,539644
5	20,41589	0,594345
6	7,513823	0,721573
7	0,622885	0,952059
8	0,176407	0,991395
9	0,096669	0,995698
10	0,066585	0,995698

Table 2. Classification Report

Parameter	Precision	Recall	F1 Score	Support
Label Happy	1.00	0.99	0.99	505
Label Sad	0.98	0.99	0.99	310
Accuracy			0.99	815
Macro Avg	0.99	0.99	0.99	815
Weighted Avg	0.99	0.99	0.99	815

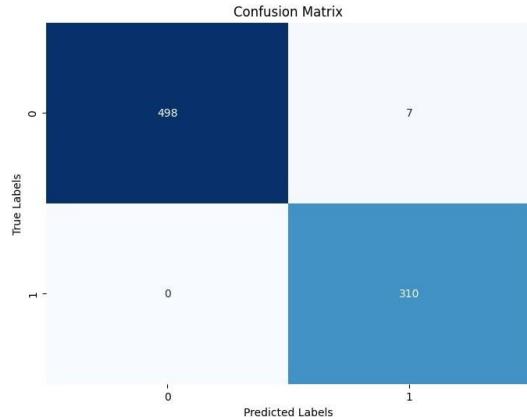


Figure 9. Matrik Confusion

Evaluate the model's performance on validation data to check the extent to which the model is working on never-before-seen data. Evaluation metrics such as accuracy and loss will be used to measure model performance. Visualize experimental results such as training loss charts and accuracy during the training process. This visualization helps understand the performance of the model and identify whether the model is overfitting or underfitting. As the author has done in explaining the results of the study using graphs that have been presented in the previous explanation as in figure 6, figure 7, and figure 8.

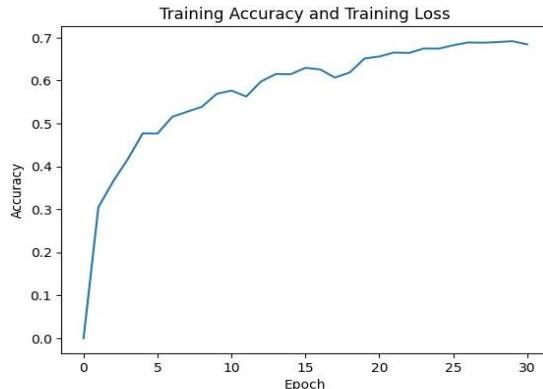
Analysis of experimental results is performed to understand model performance and identify steps that need to be taken to improve model performance. The experiment flow above is our experiment flow in classifying images using TensorFlow.

Comparison with Pytorch

In addition to training on TensorFlow, the author also compared with other models using pytorch in training models to add new insights related to deep learning training models and also to find out how powerful the model is.

The steps and parameters that the author uses are the same as the parameters used when training a model using TensorFlow. The results compared are the results of val loss, val accuracy, and training time. Here are the accuracy results obtained when using pytorch.

Figure 10. Pytorch Accuracy Plot



Based on figure 10 plot accuracy pytorch experienced a good training trend, where when the first epoch to the fifth experienced an increase from 0 to 50%, when training to the sixth to ten epochs the model still increased even though the increase was only about 10%, and when the tenth to fifteenth training models began to increase and the decrease in accuracy was not significant, and experienced a model increase of about 3% from the previous one, for the results of the fifteenth epoch to the last epoch yielded the highest accuracy at 69%. For loss results in pytorch training will be explained in the picture below.

Based on Figure 11 the pytorch loss plot experienced a good training trend, where when the first epoch to the tenth experienced a decrease in loss from 1.741649 to 1.132407 when training to the eleventh to twenty epochs the model still decreased to 0.944578, and the loss. In the last epoch, it yields a value of 0.890708. The more detailed results of the training are in the form of a table below.

And the results of the comparison of the two models will be shown in more detail in the following table.

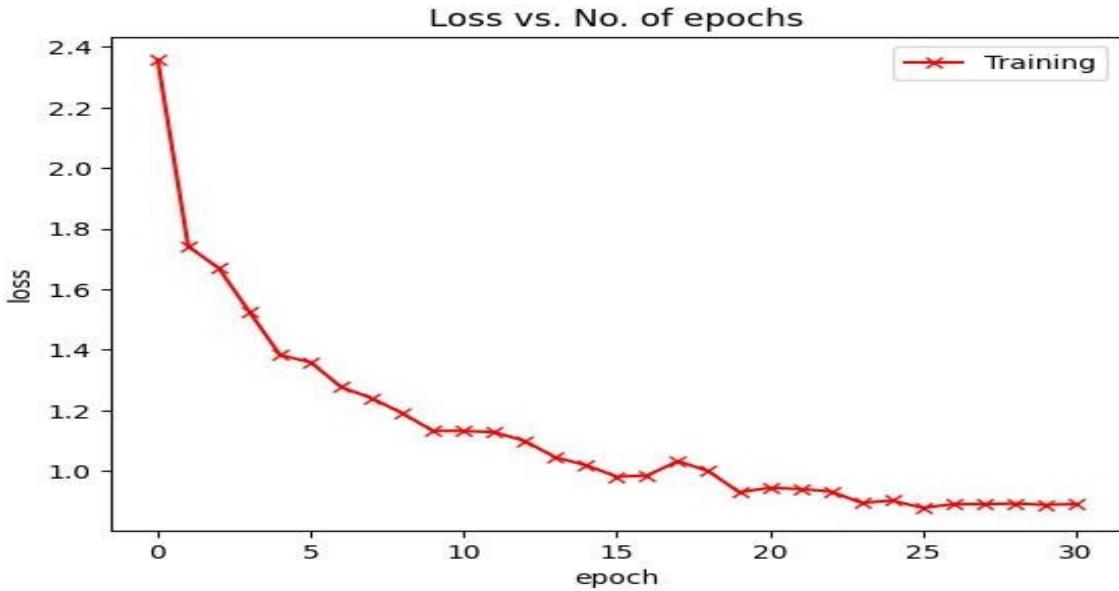


Figure 11 Results of Loss Training Using Pytorch

Epoch	Tensor Flow		Pytorch	
	loss	accuracy	loss	accuracy
1	477,9232	0,37984	1,741649	0,304899
2	6658,777	0,37984	1,669129	0,365145
3	1033,546	0,37984	1,524892	0,417456
4	83,63907	0,385986	1,382684	0,476824
5	7,269608	0,587584	1,35904	0,47642
6	1,276573	0,850031	1,277102	0,515473
7	0,134068	0,978488	1,239992	0,527222
8	0,060849	0,989551	1,190474	0,538687
9	0,043518	0,991395	1,132247	0,568889
10	0,035877	0,993854	1,132407	0,576528
11	0,03148	0,993854	1,128084	0,562582
12	0,028882	0,993854	1,09911	0,597708
13	0,030343	0,992624	1,044825	0,615069
14	0,029766	0,99201	1,019118	0,614489
15	0,0292	0,99201	0,981262	0,629362

Table 4. Comparison of TensorFlow and Pytorch

Based on the comparison results in the table above, it can be seen that TensorFlow produces better accuracy, where the highest accuracy in TensorFlow is 99% and Pytorch is only 69%, there is a difference of 24%. For loss accuracy, TensorFlow is better with the lowest loss result at 0.196418 while on the pytorch only at 0.878375 there is a considerable difference at 0.683757. In terms of training time on TensorFlow for 3 hours 2 minutes 55 seconds, and pytorch for 16 minutes 11 seconds.

Advantages of Using face recognition:

- Detection Efficiency: By leveraging the face recognition library, the face detection process becomes faster and more efficient. The HOG method used has been optimized for the best performance.
- Improved Accuracy: Good face detection is a critical step in recognizing emotional expressions. By using a tested library, the accuracy of face detection can be enhanced.
- Data Preparation: The use of bounding boxes and cropping assists in preparing more focused and relevant data for further analysis.
- Interoperability: The face recognition library is integrated with OpenCV and can be used in various stages of image analysis, including emotional expression detection.

The use of the face recognition library as the first step in image analysis can significantly improve the quality of data passed to the CNN model for emotional expression recognition. The author also implemented the created model to detect facial expressions at several strategic locations to test the model and for the implementation of calculating the happiness index in different places. For the model, the author chose TensorFlow because it performed better in

terms of accuracy, loss, and training time compared to PyTorch. The author conducted implementation examples in four different locations.

Conclusion

From the results of the research conducted using TensorFlow and PyTorch models for emotion detection based on facial expressions, it can be concluded that the model's performance is greatly influenced by the choice of architecture, hyperparameters, and the quality of the data used. A well-configured Convolutional Neural Network (CNN) model empowered by TensorFlow technology has demonstrated a higher level of accuracy compared to the model developed using PyTorch. The highest performance achieved by the TensorFlow model is 93% in recognizing emotional expressions, while the PyTorch model reached an accuracy of 69%. Additionally, the TensorFlow model also produced lower loss accuracy and had a shorter training time compared to the PyTorch model.

This research emphasizes the importance of selecting the appropriate platform and technology for model development. TensorFlow, with its optimal support for tasks such as image processing, has proven its effectiveness in supporting facial expression recognition based on human faces.

Based on these findings, in the context of calculating the happiness index in a Smart City, the choice of the right technology can significantly impact the accuracy and efficiency of measurements. The TensorFlow platform, which has demonstrated better performance in this research, can be a strategic choice for integrating facial expression detection technology in calculating the happiness index in a location.

This conclusion contributes to efforts to optimize facial expression detection technology to support Smart City goals in creating a more responsive environment that meets the emotional needs of its residents. By combining this research with relevant social, environmental, and demographic data, Smart Cities can more accurately measure the happiness index and take appropriate steps to improve the emotional well-being of their population.

References

- Arrohman, Z. D., & Andriani, W. (2022). *Application of Artificial Intelligence In Smart City Development*. *International Journal of Engineering Business and Social Science*, 1(01), 11–16.
- Assiri, B., & Hossain, M. A. (2023). Face emotion recognition based on infrared thermal imagery by applying machine learning and parallelism. *Mathematical Biosciences and Engineering, MBE*, 20(1), 913–929.
- Athavle, M., Mudale, D., Shrivastav, U., & Gupta, M. (2021). Music recommendation based on face emotion recognition. *Journal of Informatics Electrical and Electronics Engineering (JIEEE)*, 2(2), 1–11.
- Bégue, I., Vaessen, M., Hofmeister, J., Pereira, M., & emotion recognition. *Future Internet*, 14(1), 5.
- Grahlow, M., Rupp, C. I., & Dernit, B. (2022). The impact of face masks on emotion recognition performance and perception of threat. *PLoS One*, 17(2), e0262840.
- [1] G. Zhao and M. Pietikäinen, “Dynamic texture recognition using local binary patterns with an application to facial expressions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, 2007.
- [2] C. Shan, S. Gong, and P. W. McOwan, “Facial expression recognition based on Local Binary Patterns: A comprehensive study,” *Image and Vision Computing*, vol. 27, no. 6, pp. 803–816, 2009.
- [3] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.
- [4] Y. Tang, “Deep learning using linear support vector machines,” *arXiv preprint arXiv:1306.0239*, 2013.
- [5] A. Mollahosseini, D. Chan, and M. H. Mahoor, “Going deeper in facial expression recognition using deep neural networks,” *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.
- [6] S. Minaee, A. Abdolrashidi, and Y. Wang, “Deepemotion: Facial expression recognition using attentional convolutional network,” *Sensors*, vol. 21, no. 9, p. 3046, 2021.
- [7] P. Khaire and A. Kumar, “Comparative analysis of Schwartz, S., & Vuilleumier, P. (2019). Confidence of emotion expression recognition recruits brain regions outside the face perception network. *Social Cognitive and Affective Neuroscience*, 14(1), 81–95.
- Cai, W., Gao, M., Liu, R., & Mao, J. (2021). MIFAD-net: multi-layer interactive feature fusion network with angular distance loss for face emotion recognition. *Frontiers in Psychology*, 12, 762795.
- Ferreira, P. M., Marques, F., Cardoso, J. S., & Rebelo, A. (2018). Physiological inspired deep neural networks for emotion recognition. *IEEE Access*, 6, 53930–53943.
- Gloor, P. A., Fronzetti Colladon, A., Altuntas, E., Cetinkaya, C., Kaiser, M. F., Ripperger, L., & Schaefer, T. (2021). Your face mirrors your deepest beliefs—Predicting personality and morals through facial CNN and hybrid models for facial emotion recognition,” *Procedia Computer Science*, vol. 167, pp. 2312–2321, 2020.
- [8] S. Li, W. Deng, and J. Du, “Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition,” *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 356–370, 2019.
- [9] S. L. Happy and A. Routray, “Automatic facial expression recognition using features of salient facial patches,” *IEEE Transactions on Affective Computing*, vol. 6, no. 1, pp. 1–12, 2015.
- [10] B. Hasani and M. H. Mahoor, “Facial expression recognition using enhanced deep 3D convolutional neural networks,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.
- [11] A. Majumder, R. Dey, and P. Bhowmick, “Real-time emotion recognition from facial expressions using deep CNN and OpenCV,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 10, no. 6, 2021.
- [12] R. Kumar, N. R. Singh, and P. Gupta, “Design of a real-time facial emotion detection system using deep learning and Tkinter,” *International Journal of Computer Applications*, vol. 183, no. 15, pp. 12–18, 2022

