Vol: 06 Issue: 06

International Journal of Data Science and Advanced Analytics (ISSN: 2563-4429)

Detecting Facial Emotion Expression to Mentor and Manage Crowds Using a Deep Convolutional Neural Network

Ratna Bandaru [1], Walaa Bajnaid[2]
[1] Carrier, Indiana, US, [2]King Abdulaziz University
[1] Ratnakumar.Bandaru@carrier.com, [2]wnbajnaid@kau.edu.sa

Corresponding author's email: wnbajnaid@kau.edu.sa

***Abstract*—**Human interactions are heavily reliant on facial expressions. An individual's emotional state can affect their safety and security in crowd places. Thus, monitoring and detecting emotions in facial expression in real time contributes to preventing potentially harmful situations. However, performing such tasks is difficult and complex and requires advanced computational methods. This study aimed to use a deep convolution neural network to detect and monitor facial emotion expressions in crowds. In addition, the effectiveness of transfer learning using VGG16, ResNet50 and Xception with DCNN models to improve accuracy was investigated. To achieve this, an FER2013 dataset with more than 35,000 images classified in terms of anger, fear, disgust, happiness, surprise, sadness and neutral was used. The results showed that transfer learning improved the accuracy and performance of an ensemble of the three models.

***Keywords*—Crowd management, deep learning convolutional network, emotion deduction, facial expression.**

1. Introduction

Crowd management is a complex task that requires effective planning, infrastructure and emergency services to ensure publicity safety [1]. In crowds, emotion forms a crucial part of human interaction and a response system to surrounding events [2]. However, in dangerous situations, the human brain mechanically scans the environment for threats and sends alerts to the body to handle the threats [3]. Facial expressions reflect a person's emotions. The human face conveys a wide range of emotions, which vary from mild to intense [3]. Fear, anger and anxiety reveal unconvinced emotions. For example, expressions of anger are explained as 'back away', and identifying facial emotions in crowds benefits both crowd and social security [3]. Analysing emotions assists in detecting abnormal and suspicious behaviour that threatens security. The results from processed crowd data that are acquired during event execution guide decision-makers to avoid crowd disasters.

Facial emotion identification has been studied for over 20 years and remains a significant topic, especially following technological advancements. Crowd-monitoring technology is imperative for identifying potentially critical situations. Thus, developing techniques to detect facial expressions and recognise emotions has generated increased interest. However, the development of tools for recognising emotions remains controversial and challenging regarding the accuracy of these tools and the complexity of interpreting expressions across cultures [4]. In addition, technology such as CCTV is still not a very effective method due to its dependence on human factors. A person watching a camera may miss surveillance footage [5]. A crowd environment, such as a public event or transportation, is also a big challenge for emotion detection tools. Distractions from crowded environments affect the ability of emotion recognition tools.

Facial emotion recognition is an advanced system that uses a combination of machine learning and computer vision techniques to identify real-time facial expressions and is associated with video surveillance [6]. Several applications have been developed for managing crowds and maintaining security. For example, a facial recognition application was used by the Brondby football club in Denmark to recognise fans with a history of disorderly behaviour to ban them from attending games [7]. However, most security surveillance applications have used deep learning methods [8]. On the other hand, the optical flow logarithm has been used to analyse images with impregnated noise [2].

Deep convolutional neural networks (DCNNs) are a popular image classification method that has been used for facial expression recognition. This method is characterised by its accuracy and performance [9]. Background noise, occlusions and lighting affect the performance of image classification and facial detection systems. In addition, accuracy and timing are two critical factors when using facial expression identification in the context of crowd management and monitoring. Thus, this study seeks to examine the use of DCNNs in accurately detecting emotions from facial expressions in crowds. To achieve this aim, we build a system that identifies emotions in crowds in the real word by detecting facial expressions using deep learning and big datasets containing facial images.

2. Research Background

2.1. Crowd Analysis

A crowd is an assembly of a large number of people in a place where they can communicate and respond to stimuli in similar ways [10]. Mass events are influenced by several factors, such as crowd size, belonging, anonymity and impunity. The fact that emotions and destructive behaviour are contagious in crowds may cause people to participate in irrational collective actions, become bolder and lack inhibition [11]. Emergency and security management in mass events is a critical task that requires real-time responses to abnormal actions and quick decisions [5]. Therefore, automated methods are necessary for monitoring crowds, maintaining safety and enhancing customer experiences in public places, such as public transportation, banks and shopping malls [8].

Crowd behaviour analysis is a technology that sends alerts when anomalies are detected in crowds [12]. The mechanism of these technologies relies on real-time footage provided by CCTV in locations where gatherings take place [12]. The huge amount of video data generated from CCTV needs to be processed in an efficient and effective way. Therefore, smart surveillance with wireless and robotic sensors has replaced traditional methods of predicting abnormal behaviour and does not require human intervention [13].

2.2. Facial Expressions

Emotions are mental and psychological experiences triggered by specific events and objects. They involve conscious feelings that range between positive, negative and neutral. People use emotions to communicate, interact and navigate for their social and physical environments. Based on the study by Ekman (2011), seven basic categories of emotions are expressed in the human face: anxiety, surprise, anger, sadness, disgust, happiness and neutral. These emotions are considered universal despite differences in intensity and expression between cultures [14].

Facial expression is the most frequently used form of non-verbal communication. Therefore, any rapid change in facial expression from one emotion to another indicates abnormal behaviour. Likewise, a face expressing a high intensity of emotions could show signs of an emergency through the inconsistency of their expressions or the contradictions in the emotions that appear in different parts of the face [6].

2.3. Facial Emotion Recognition Systems

Facial emotion recognition (FER) is a technology that uses machine learning and computer vision to identify emotions expressed in videos or images in real-world situations [15]. The system is trained to recognise emotions from facial expressions through the classification of images into these emotions.

A FER system uses two approaches: holistic and analytical. While the former deals with the entire face or the appearance of the face, the latter is concerned with geometric facial features or facial defamations, such as of the eyebrows, nose and eyes [15]. The system works through three stages: face detection, the extraction of facial features and the classification of facial expressions [9].

2.4 Challenges of Facial Emotion Recognition

FER faces three main challenges—intra-subject variability, inter-subject variability and ambiguity—leading to complexity. At the inter-subject level of variability, the same emotion can be expressed differently among individuals. However, intra-subject variability the same person can evoke the same emotional state through different facial expressions. Ambiguity refers to when the facial expression represents more than a single emotion . In addition, the accuracy of recognising emotions associated with each movement of facial muscles and features is a challenge, as is the intensity of these emotions. Cultural social factors, as well as the context of the scene affect emotional expression.

Changes in illumination, attitude and location affect FER performance, as follows:

Lighting**:** This greatly impacts FER performance, with poor lighting, the illumination conditions and the location (e.g. indoors or outdoors) affecting the brightness or darkness of a photo, which may cause poorer recognition of facial expressions.

Face pose and angle: The orientation of a face towards the camera plays an important role in recognising the correct facial expression. The same expression could indicate different emotions from different angles. FER training uses photos with a zero or minimal angle of face orientation in a frontal position. However, in a crowd, some facial features are obscured, resulting in incorrect recognition. Emotions can be expressed with a high degree of intensity or subtly.

Occultation: Any covering of the face, either full or partial, makes it hard for a system to recognise the face. Objects such as sunglasses or scarves and overlapping other faces in the image could impact facial expression identification. Nonetheless, many attempts have been made to overcome these obstacles by segmenting the occluded images.

Face variation**:** Generally, FER is trained on neutral faces. However, if a face image is irregular, such as with a moustache, eyewear, a smile or a frown, this will affect the facial classification. This challenge can be overcome by training the logarithm to deal only with optimal features [3].

Interclass similarities**:** This refers to identifying people with highly similar facial characteristics.

Skin colour and ethnicity**:** Some FER systems recognise some ethnicities and skin colour better than others.

Face size and resolution**:** FER captures larger faces and photos with a high resolution more accurately.

Dataset size**:** A system trained on a large dataset with diverse facial expressions and demographics performs better [16].

2.5 Comparing FER Between Deep Learning and Conventional Learning Approaches

In general, the conventional approach is the old method of recognising facial expressions using hand-crafted features. This method has improved over time. However, it still has many limitations. In contrast, the deep learning method has the ability to produce better results with accuracy, stability and robustness while dealing with high-volume and complex data. Li and Deng [17] showed that FER performed with deep learning achieved outstanding results in comparison to conventional methods.

2.5.1 Conventional FER Approach

The conventional FER approach is a method of recognising facial expressions using a traditional computer technique over several stages. This approach begins with the image pre-processing step, which includes two sub-steps: face detection and facial extraction. The face detection step aims to identify a face in an image and extract its regions. Several techniques, such as hear cascades, a histogram of oriented gradients, augmentation, normalisation and face alignment, may be included in the pre-processing of an image [18]. This is an essential step in improving accuracy. The facial feature extraction step involves identifying important features of the face, such as the eyes, nose, eyebrows and mouth, using geometrics and the texture and intensity of facial features. The second step is the classification of expressions, which determines emotions following the extraction of facial features. This can be done using several techniques and algorithms, such as artificial neural network, decision tree, K-nearest neighbour and support vector machine. However, this approach has limitations in terms of computational cost, difficulty in generalising results and limited scalability. Due to its handcrafted hyperparameters and manual settings, this approach is time-consuming and error-prone and has low recognition [19].

2.5.2 FER Using Deep Learning

Deep learning is a type of machine learning that uses artificial neural networks to predict features and patterns. This approach has the ability to deal with large amounts of data and complex emotions or facial features. In comparison to conventional FER methods, deep learning–based FER is able to learn and extract features automatically from row data, which results in robust, accurate and generalisable results. There are multiple methods of deep learning that can be used to recognise facial emotions, including convolutional neural networks (CNNs), generative adversarial networks, recurrent neural networks and transfer learning.

2.5.3 Convolutional Neural Networks

CNNs are deep learning artificial neural networks used for image classification. They consist of three deamination layers: convolution, pooling and fully connected layers [20]. The output for each layer is an input for the subsequent layer by convolution and pooling. CNNs automatically learn and extract features using the algorithm of backpropagation [21]. CNN architecture is influenced by visual perception, and CNN kernels form receptors that simulate the function of neural electric singles when they exceed the threshold and reach the next neuron. Thus, CNN is able to expact and optimise loss funcatuion [22]. Facial emotion recognition has been developed and deployed using a range of CNN architectures, including LeNet, AlexNet, VGGNet, ResNet and InceptionNet.

3. Materials and Methods

FER201, a pre-existing dataset for facial emotions, was used in this study. Available on Kaggle and created by Pierre-Luc Carrier and Aaron Courville, this dataset contains 35,887 grayscale images sized at 48 × 48 pixels. Seven different emotions can be represented in each image: disgust, anger, sadness, fear, suprise, happiness and neutral. The images were originally collected from YouTube, movies and the web. A human classified each image to assign an emotion to each image see figure 1.

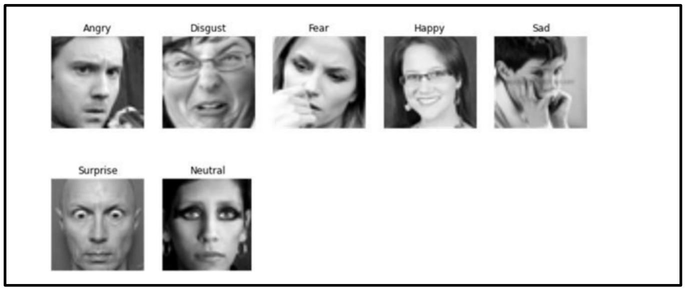

Figure 1Sample images from FER2013 dataset

3.1. Pre-processing of Data

This step is crucial for ensuring the performance and improving the accuracy of the model. In this step, the data in the form of images or videos will be processed by correcting the image and reducing the nose, extraction features and standardisation. In addition, the variation addresses illumination, head poses and different backgrounds. Normalisation data allows the information of facial expression in the images to transfer. Then, the image will pass through noise reduction to remove noise, such as salt and pepper n and Gaussian noise. Afterwards, the image correction step which involves resizing, reorientation and changing the image to grayscale. Next, is the normalisation of the image and keeping them in standard size as well as scaling the intensity values. The fourth step involves data augmentation, which includes rotating, flipping, translating and scaling the image. The final step is preparing the data by dividing it into validation, training and test sets.

3.2. Models

In this research, the Ensemble CNN model has used as a bassline model combined with other techniques of hyperparameters, guided backpropagation and optimisation. This selection of methods is based on performance and accuracy. The ensemble CNNs accuracy scored 75.8% based on the study of Khanzada et al. [23], hyperparameter optimisation accuracy scored 72.16% [23] and guided backpropagation scored 66 %in accuracy [24]. This combination of techniques is assumed to prove a more effective performance.

3.4 Transfer Learning

In this stage, the model is prepared to form in with new data and learn to extract features. The issues of the FER2013 dataset, such as small size and class imbalance, are also addressed.

3.4.1 Pre-trained models on a small dataset are adjusted to minimise the labelled data and improve performance. The process is as follows: the pre-trained model is selected based on its similarity to the current task. Then, fine-tuning is performed, which involves modifying the pre-trained model by deleting the last layers and adding new ones. Finally, the modified model trains on the new dataset and updates the weights of the model's performance. The pre-trained step results in better performance and saves on computational resources and time.

3.4.2 Image classification using visual geometry group-16 (VGG-16) architecture is designed with 16 layers, 13 of which are convolutional layers and three are connected layers. The convolutional layers are made up of 3 × 3 kernels, with a one-pixel stride, while the pooling layers have kernels measuring 2 × 2, with a two-pixel stride. There are three fully connected layers, with the last layer being the output layer (see Figure2. Consequently, the classes have a probability distribution. There are more than 138 million training parameters in the VGG-16 network, and accurate image classification has been accomplished.

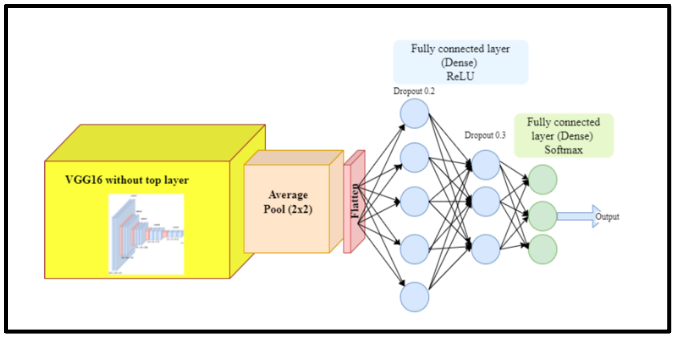

Figure 2 VGG16 network architecture

The input is 224 × 224 × 3 RGB images layered into two layers each with 64 filters, 3 × 3 kernel size and ReLU activation function. Then the Max pooling with 2 stride and 2×2 kernel. The input passes into two convolutional layers, with 128 filters and 3 × 3 kernel. Afterwards it goes into max pooling exactly, like the first one. After this, the input passes through three layers of 256 filters, then another six layers of 512 filters, and between each three layers is the max pooling layer. The results then pass into a flattened layer, followed by a fully connected layer that has 4096 units and a ReLU activation function. The subsequent dropout layer has a rate of 0.5. Next, the data flow into fully connected layers, followed by the dropout layer. Finally, in the output layer, the SoftMax activation function generates the probability distribution.

3.4.3 A convolutional network architecture residual network 50 (ResNet50) is used to address the vanishing gradients that occur in a deep neural network. ResNet50 is a residual block network made up of 50 layers that allows the convolutional layers to be bypassed and adds the input directly to the output of the network. ResNet50 is pretrained on 1000 different classes on over 1 million images (figure 3).

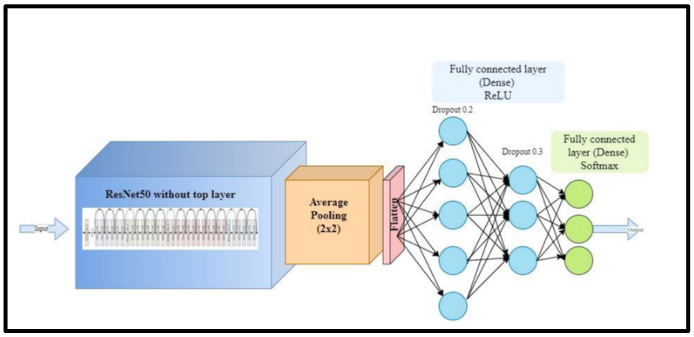
Fully connected layer (Dense) ReLU
Dropout 0.2
Dropout 0.3
Fully connected layer (Dense) Softmax
Input
ResNet50 without top layer
Average Pooling (2x2)
Flatten
Output

Figure 3 ResNet50 network architecture

ResNet50 processes data in a convolutional layer that has 64 filters and a size of 7 × 7 kernels, with two strides and three paddining. Then, the normalisation layer is flowed by the ReLU activation function. After this is the max pooling layer, with a 3 × 3 pool size and two strides. Next, there are four stages that have residual blocks. In the first stage, there are three residual blocks with 128 filters each. The number of residual blocks and filters rise in the second and third stages, respectively, to four and s residual blocks, with 128 and 256 filters in each block. The residual blocks are reduced to three in the fourth stage, while the filters are 512 in each block. Next is the average pooling layer, where the feature maps are computed for each spatial dimension. Finally, the SoftMax activation function generates the probability distribution in a layer of 1000 nodes.

3.4.4 Deep convolutional network Xception network architecture performs image classification tasks, with several convolutional block layers followed by a connected layer. A series of depthwise separables forms each convolution layer. The benefit of depthwise separable convolution is to minimise the computation and parameters required while keeping the expressive power at the same level. This is due to the conversion of the standard convolution into a depthwise and pointwise convolution.

The Xception model has a depthwise separable convolution that contains eight filters with a 3 × 3 kernel size. Then comes the normalisation layer and the activation of the ReLU layer, followed by a max pooling layer with a 2 × 2 pool size and convolution blocks of two depthwise separables of 16, 32 and 64 filters, each with 3 × 3 kernels. All blocks have ReLU activation and normalisation layers. The last block has a max pooling layer with a 2 × 2 pool size see figure 4.

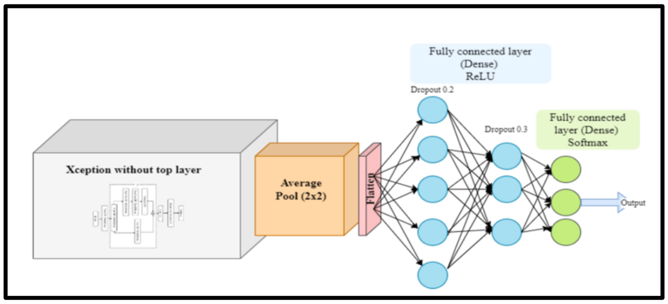
Fully connected layer (Dense) ReLU
Dropout 0.2
Dropout 0.3
Fully connected layer (Dense) Softmax
Xception without top layer
Average Pool (2x2)
Flatten
Output

Figure 4 Xception network architecture

3.4.5. Ensemble Network Architecture

This model consists of three transferred learned models: VGG16, Res Net50 and Xception. The model contains two parts. The first part is an ensemble part formed by adding three models of VGG16, Res Net50 and Xception, and the second part involves fully connected (FC) layers in which the predictions of the models are combined and output and passed into the SoftMax layer to obtain the final predictions. Then, for each class, the weight of the total prediction is converted to probabilities see figure 5

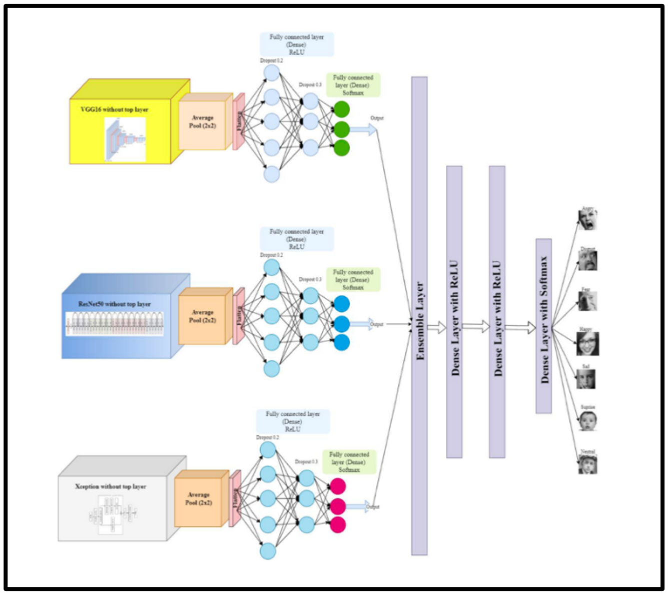

Figure 5 Ensemble network

3.5. Hyperparameter Tuning and Model Training

The prepared dataset is used to train the model in this stage. In addition, the hypermeters are adjusted. As part of this process, the hyperparameters are defined. This includes a loss function, an optimiser, a learning rate, a batch size and an epoch number as follows:

Learning rate: A low or high learning rate may result in slow convergence or slow learning. Optimising the Xception network requires tuning the learning rate.

Batch size: During one iteration of the training algorithm, the batch size determines how many samples are processed. A small batch size causes noise in optimisation, while a large batch size causes slower convergence. The optimal size results in stable and faster training.

Number of epochs: This refers to the number of training networks in the entire dataset. An incorrect number of training times affects the fitting. Thus, underfitting occurs when there are too few epochs, while overfitting occurs when there are too many epochs. Adjusting the epoch number leads to determining the optimal point.

Dropout rate: This is a technique that randomly drops out some neurons during training to prevent overfitting and to improve performance.

Optimiser: This algorithm is responsible for updating network parameters during the training process. The optimiser affects the performance and convergence speed. NAdam is a commonly used optimiser.

Weight decay: This is a regularisation technique that works by adding penalties to the loss function and aims to encourage networks to have smaller weights. This hyperparameter maintains balance between overfitting and underfitting.

Early stopping: This is an early intervention to prevent overfitting during the training process. These hypermaters should be tuned to the patience parameter, which refers to how long to wait for improvement before stopping the training, if none are observed.

3.6. Techniques for Model Evaluation

It is important to prove the method's strength using evaluation metrics. Thus, several evaluation metrics can be used with an FER, including the following:

3.61 Confusion matrix, which evaluates classification algorithm performance by comparing the actual labels with the predicted ones. This matrix represents four confusion types: TP (true positive), TN (true negative), FP (false positive) and FN (false negative). ***Accuracy*** is the number of true predictions divided by the total number of all correct and incorrect predictions: Accuracy = (TP + TN) / (TP + TN + FP + FN). ***Precision*** calculates the percentage of TP/total positive predictions. ***Recall*** is measured by the percentage of TP/total actual positives. The ***F1-score*** combines precision and recall to provide a balanced measure of model performance 2 × ((Precision × Recall) / (Precision + Recall)). ***Sensitivity*** determines the model's ability to identify true positive emotion and is calculated by finding the ratio of TP to (TP + FN). ***Specificity*** determines the model's ability to identify true negative emotions and is calculated by finding the ratio of TN to (TN + FN).

C. Design and Analysis

In this study, exploratory data analysis is used to understand the characteristics and select the augmented data method. As mentioned above, FER2013 is used as the dataset. However, the classification of emotions in this dataset is as follows: 17% neutral, 25% happy and 11% surprised (see Figure). The images with natural emotions posed a challenge in identifying the emotional features.

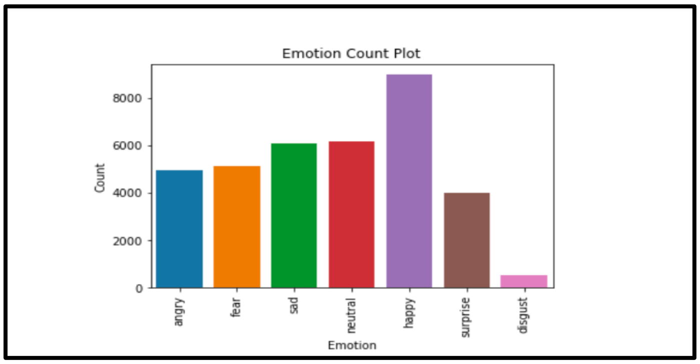

Figure 6 Class distribution in dataset

3.7. Data Preparation and Cleaning

To avoid bias caused by overrepresenting, 1853 duplicated images were removed. Then, the data were normalised by extracting and labelling the images and converting them to a NumPy array. Thus, the mean and standard deviation were calculated to determine that the range of images was between 0 and 1.

3.7. Augmenting the Data

The aims of this step were to reduce overfitting, improve generalisation and capture the underlying distribution of the data. In this study, rotations, horizontal mirroring, image zooms and both horizontal and vertical shifting were the augmentation techniques used. Keras data generators were used with these techniques to automatically resize and format the images. Then, using real-time data augmentation, Python's ImageDataGenerator class generated tensor image batches. In the constructor of the ImageDataGenerator, the data augmentation techniques applied the following parameters:

- **Image sizes** were fixed at 224 × 224 for VGG16 and ResNet and 229 × 229 for Xception.
- The **image rotation** was set to 10 degrees and ranged from 0 to 180 degrees (randomly rotating the images).
- Horizontal and vertical **image shift** was set to a range of 0.1 (randomly).
- Image **zoom in and out** were set to a range of 0.1 (randomly).
- **The horizontal flipping** of the image was set to True.
- **Rescaling** was set to 1/225, with a range of 0,1.

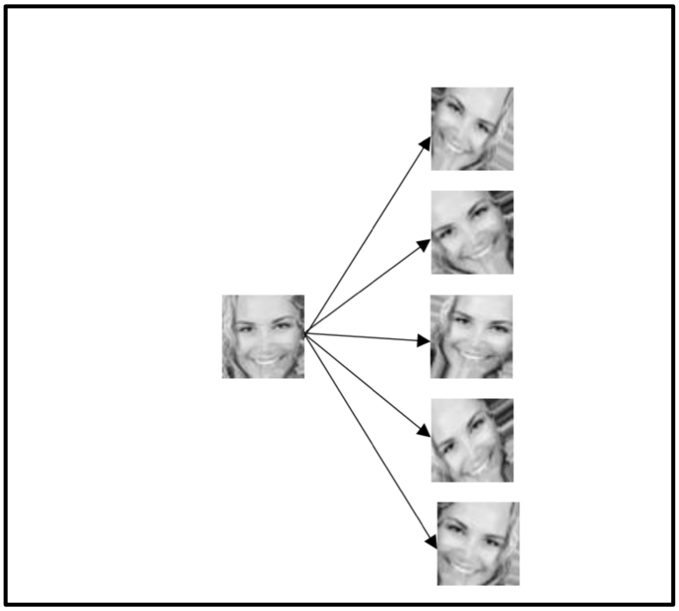

Figure 7Data augmented with

3.8. Partitioning Data

The FER2013 dataset is classified into three sets. The training set contains 28,709 images and was used to train the model. The validation set contains 3589 images and was used to monitor the model's performance and adjust the hyperparameters. Lastly, the testing set contains 3589 images and was used to evaluate the model's performance. There are four columns in the dataset: pixels, emotion label and usage type (which identify the type of the dataset to which the image belongs).

3.9. Implementation of the Model

The pre-trained models accept the input of 224 × 224 RGB images in 48 × 48 grayscale. Therefore, during training, the research dataset was recoloured and resized. Kim et al.'s [25] model was adopted for this study. The model contains three stages: convolution, max pooling layers and FC layer. The third stage is the SoftMax output layer.

- The convolution layer has 32, 32 and 64 filters with sizes of 5 × 5, 4 × 4 and 5 × 5.
- The max pooling layer has 3 × 3 kernels in a size of 3 × 3 and two strides and a ReLU activation function.
- At every layer, batch normalisation was added, and 30% dropout was applied after the last layer of FC.
- One hundred epochs were used to train the model, and 0.9 momentum stochastic gradient descent was used to optimise cross-entropy loss.
- The initial learning rate was fixed at 0.1 and reduced to half if the accuracy did not improve for 10 epoch batch sizes set to 128 and with the weight adjusted to 0.0001.

3.9.1 VGG16 Model

The implementation of this model included the following steps: importing models from Keres; adjusting the shape of input; creating a VGG16 model using pre-trained weights and freezing the layer, loading and pre-processing the dataset; and generating batches using ImageDataGenrator.flow_from_directory(). Using model.complie() enables the model to be compiled with a specified loss function, optimiser and metric; using model.fit() enables it to train the model and determine the number of epochs; using model.fit() enables it to evaluate the model; and using model.save() enables it to save the model.

3.9.2 ResNET50 Model

In this model, the output layer was replaced by two FC layers, one with a size of 1096 and the other with a size of 1024, and a SoftMax layer with seven emotions. The first 170 layers were frozen to speed up the process. Optimising stochastic gradient descent (SGD) was used with a 32 batch size and a 0.01 learning rate. An accuracy of 73.2% was achieved as a result of the 122 epochs. It is important to note that the model did not fit the training set after various adjustments of the hyperparameters and freezing of the pre-trained network.

3.9.3 Xception Model

A deep learning framework was used to implement the model. In addition, the hyperparameters of the model were selected using grid or random search techniques. The performance of neural networks improves using normalisation techniques; however, hyperparameter tuning was used to determine the search space. In our research, the search spaces were as follows: 0.001, 0.01 and 0.1 were the learning rates; 0.1, 0.2 and 0.3 were the dropout rates; 16, 32 and 64 were the number of filters in convolutional layer number one; and 32, 64 and 128 were the number of filters in convolutional layer number two.

To evaluate the combination of hyperparameters, a grid search was applied using the Keras Tuner library. Input hyperparameters were used to create the Xception network, which was then trained on the FER data. To prevent overfitting, an early stop was used. In the validation set, the hyperparameters with the best performance were selected. To train the model, the following values were obtained using a grid search: 32 batch size, 100 epochs, 5 stop patience, 2 reduce patience rate and NAdam optimiser.

3.9.4 Ensemble Network

As explained earlier, this technique creates a new model by integrating several models to achieve better performance and accuracy. The process of integration passes through two stages: ensembling of the models and fully connecting the layers. To attain the final predications, the output went onto the SoftMax layer. The model was trained on FER data, and its performance was evaluated using a separate test. This model used the same hyperparameters as in the individual model.

4. Results

To recognise facial emotions, an ensemble of transfer training networks was applied using three networks model: VGG16, ResNet and Xception. The results showed that the transfer learning model performance using VGG16 with regard to accuracy ranged between 0.7767 and 0.4889 at the lowest accuracy. This means that the model correctly classified 77.67% of the images. In contrast, more than half of the data may have been misclassified at the lowest level of accuracy. However, validation accuracy was 0.703, which indicates that 70.3% of the images were classified correctly in the validation dataset. The lowest value was 0.5738, which means that the model performed poorly on the validation dataset. These results may indicate that the model was overfitting.

The results showed that using ResNet50 in the transfer learning model was more effective due to the accuracy increasing and the loss decreasing in the training epochs. A steady increase in accuracy occurred from the first epoch (40.56%) to the last epoch (81.66%). On the other hand, the loss decreased from 3.03 to 0.4856 between the first and last epochs. This finding suggests that, on the validation dataset, the model was able to generalise. The model was also able to recognise differences between emotions. It is important to note that after 20 epochs, both training and validation accuracy plateaus were reached, which means that any additional training would not improve the performance.

The performance of the model using Xceotion was effective, as shown in the research results. The accuracy increase during the training process reached 73.9%. This result indicates that the model performed effectively on the validation dataset and was able to generalise to new data. The model then proved that it was able to learn by capturing features from data and identifying relationships and patterns.

Regarding the ensemble network, the results showed that the accuracy achieved 76.2%. This score outperformed the individual models.

4.1 Results of the Evaluation Method

A comparison of the depth, parameters, epoch time and testing accuracy on the VGG16, ResNet and Xception models is shown in Table 1. The results showed that ResNet50 and Xception had better performance and fewer parameters. However, Xception consumed more time per epoch.

Table 1Networks models and their parameters results

Network	Depth (layers)	Parameters (millions)	Testing accuracy	Epoch time
VGG16	16	138	70.3%	80s
ResNet50	50	25.6	72.8%	56s
Xception	126	22.9	73.9%	90s

The result of using the precision, f1-score, recall, sensitivity and accuracy to evaluate the three CNN models are presented in Table 2. The precision results showed that Xception had a higher score in terms of identifying TP and deducing positive samples based on recall and sensitivity scores. Likewise, in detecting negative samples, Xception excelled in the two models. With regard to the F1-score, the results showed that the overall performance for Xception was better than the other two models.

Table 2 Evaluation metrics

Network	Precision	F1-score	Recall	Sensitivity	Specificity	Accuracy%
VGG16	0.69	0.72	0.74	0.74	0.66	70.3
ResNet50	0.73	0.75	0.77	0.77	0.689	72.8
Xception	0.75	0.76	0.77	0.77	0.7	73.9

4.2 Test Evaluation

The ensemble network test evaluation scored 76.2%. The confusion matrix (Figure 8) showed that the model's emotion classification performance was high, as proved by the diagonal values. Based on the matrix results, the model faced challenges in distinguishing between anger, disgust and fear. To illustrate this, from the anger sample, only 58% of images were classified as angry and 42% were mistakenly classified as neutral, fear or surprise. Likewise, from the disgust sample, only 65% were classified correctly, and the remaining ones were misclassified.

On the other hand, happy, surprise and neutral were correctly classified.

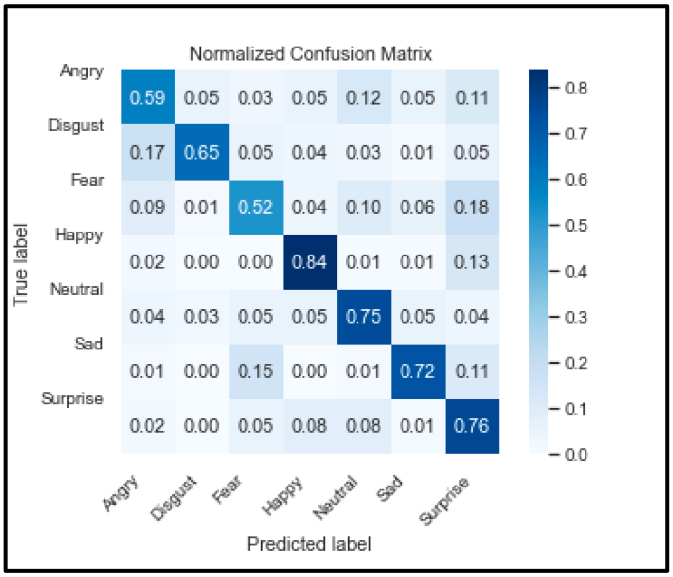

Figure 8 confusion matrix

5. Conclusion

This paper aimed to design and implement a CNN that uses transfer learning and an ensemble of deep learning for facial emotion recognition and to evaluate its performance on an FER2013 dataset in a real-world environment. Overall, the ensemble network performed better compared to using VGG16, ResNet50 and Xception models individually. In addition, the results prove that Xception has the highest accuracy and the best performance. However, the model used in the research may be biased towards some emotions, and FER2013 may not represent all populations or demographics. Thus, training the model on several datasets is important to decrease biases in populations and emotions. Further research should be conducted using other deep learning models and techniques. The results of this study prove that the model can be implemented to monitor emotional states in crowds.

References

[1] D. Sharma, A. P. Bhondekar, A. K. Shukla, and C. Ghanshyam, "A review on technological advancements in crowd management," *J. Ambient Intell. Humaniz. Comput.*, vol. 9, no. 3, pp. 485–495, 2018.

[2] Y. Miao *et al.*, "Abnormal Behavior Learning Based on Edge Computing toward a Crowd Monitoring System," *IEEE Netw.*, vol. 36, no. 3, pp. 90–96, 2022.

[3] D. V. Becker, U. S. Anderson, C. R. Mortensen, S. L. Neufeld, and R. Neel, "The face in the crowd effect unconfounded: Happy faces, not angry faces, are more efficiently detected in single- and multiple-target visual search tasks," *J. Exp. Psychol. Gen.*, vol. 140, no. 4, pp. 637–659, Nov. 2011.

[4] H. Mokayed, T. Z. Quan, L. Alkhaled, and V. Sivakumar, "Real-Time Human Detection and Counting System Using Deep Learning Computer Vision Techniques," *Artif. Intell. Appl.*, vol. 1, no. 4, pp. 221–229, Oct. 2022.

[5] M. Q. Ngo, P. D. Haghighi, and F. Burstein, "A crowd monitoring framework using emotion analysis of social media for emergency management in mass gatherings," *ACIS 2015 Proc. - 26th Australas. Conf. Inf. Syst.*, 2015.

[6] O. Kalyta, O. Barmak, P. Radiuk, and I. Krak, "Facial Emotion Recognition for Photo and Video Surveillance Based on Machine Learning and Visual Analytics," *Appl. Sci.*, vol. 13, no. 17, Sep. 2023.

[7] M. Qaraqe *et al.*, "PublicVision: A Secure Smart Surveillance System for Crowd Behavior Recognition," *IEEE Access*, vol. 12, pp. 26474–26491, 2024.

[8] K. Rezaee, S. M. Rezakhani, M. R. Khosravi, and M. K. Moghimi, "A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance," *Pers. Ubiquitous Comput.*, vol. 28, no. 1, pp. 135–151, Feb. 2024.

[9] M. Roshanzamir, R. Alizadehsani, A. Shoeibi, J. M. Gorriz, A. Khosrave, and S. Nahavandi, "What happens in Face during a facial expression? Using data mining techniques to analyze facial expression motion vectors."

[10] J. Struniawski, "CROWD MANAGEMENT DURING MASS EVENTS," *Zesz. Nauk. SGSP*, vol. 2, no. 88, pp. 27–38, Feb. 2024.

[11] W. Halboob, H. Altaheri, A. Derhab, and J. Almuhtadi, "Crowd Management Intelligence Framework: Umrah Use Case," *IEEE Access*, vol. 12, pp. 6752–6767, 2024.

[12] Sachin Bhardwaj, Apoorva Dwivedi, Ashutosh Pandey, Dr. Yusuf Perwej, and Pervez Rauf Khan, "Machine Learning-Based Crowd behavior Analysis and Forecasting," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, pp. 418–429, Jun. 2023.

[13] E. B. Varghese and S. M. Thampi, "Towards

the cognitive and psychological perspectives of crowd behaviour: a vision-based analysis," *Conn. Sci.*, vol. 0, no. 0, pp. 1–26, 2020.

[14] P. Ekman and D. Cordaro, "What is meant by calling emotions basic," *Emot. Rev.*, vol. 3, no. 4, pp. 364–370, 2011.

[15] I. Perikos, E. Ziakopoulos, and I. Hatzilygeroudis, "IFIP AICT 436 - Recognizing Emotions from Facial Expressions Using Neural Network," 2014.

[16] D. H. Lee and J. H. Yoo, "CNN Learning Strategy for Recognizing Facial Expressions," *IEEE Access*, vol. 11, pp. 70865–70872, 2023.

[17] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," *IEEE Trans. Affect. Comput.*, vol. 13, no. 3, pp. 1195–1215, 2022.

[18] M. Ubaid, M. Khalil, M. Khan, T. Saba, and A. Rehman, "Beard and Hair Detection, Segmentation and Changing Color Using Mask R-CNN," in *Proceedings of International Conference on Information Technology and Applications*, 2022, pp. 978–981.

[19] W. Hua, F. Dai, L. Huang, J. Xiong, and G. Gui, "HERO: Human Emotions Recognition for Realizing Intelligent Internet of Things," *IEEE Access*, vol. 7, pp. 24321–24332, 2019.

[20] A. Saxena, "An Introduction to Convolutional Neural Networks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, pp. 943–947, 2022.

[21] A. Patil and M. Rane, "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition," *Smart Innov. Syst. Technol.*, vol. 195, pp. 21–30, 2021.

[22] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Trans. neural networks Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, 2021.

[23] A. Khanzada, C. Bai, and F. Celepcikay, "Facial Expression Recognition with Deep Learning," *stanford University*, 2024. .

[24] O. Arriage, P. Ploger, and M. Valdenegro, "Real-Time Convolutional Neural Networks for Emotion and Gender Classification," *Procedia Comput. Sci.*, vol. 235, pp. 1429–1435, 2024.

[25] B. K. Kim, S. Y. Dong, J. Roh, G. Kim, and S. Y. Lee, "Fusing Aligned and Non-aligned Face Information for Automatic Affect Recognition in the Wild: A Deep Learning Approach," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 1499–1508, 2016.