Chapter 14
A Deep Learning Model for Visual Sentiment Analysis of Social Media

Krishna Pal Singh Tiwari, Nonita Sharma, Preeti Vats, Manik Rakhra, and Divyansh Sharma

14.1 Introduction

14.1.1 Overview

Each day social media is getting preferred that's why people are posting their everyday tasks as well as their feeling and ideas in the microblogging system like Facebook, Instagram, Flickr and also twitter consequently containing much more details of the people. Our purpose in this research study is to immediately infer favorable, neutral, and unfavorable human perspectives from pictures submitted on Flickr as well as Instagram (Ortis et al. 2020). Based upon the feeling and sentiment inferred from their images, the authors may estimate a person's psychological wellness (Ortis et al. 2020). A regular technique is to look for components in an image that are related to human feelings, such as items (e.g., toys, birthday celebration

K. P. S. Tiwari (✉)
Cognizant Technology Solutions Pvt. Ltd, Howrah, India
e-mail: krishnapal.tiwari@cognizant.com

N. Sharma · P. Vats
Department of Information Technology, Indira Gandhi Delhi Technical University for Women, Delhi 110024, India
e-mail: nonitasharma@igdtuw.ac.in

M. Rakhra
Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab 144411, India
e-mail: manik.23538@lpu.co.in

D. Sharma
Bow Valley College, Calgary, AB, Canada
e-mail: d.sharma921@mybvc.ca

N. Sharma et al. (eds.), *Big Data Analytics in Intelligent IoT and Cyber-Physical Systems*, Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-99-4518-4_15

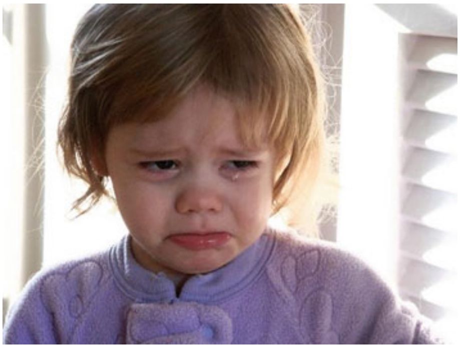


Fig. 14.1 Sobbing girl (Khorrami et al. 1510)

cakes, gun). Computer vision techniques such as face-scanning and also emotion acknowledgment can rapidly recognize the child in Fig. 14.1.

Facial emotions are important in nonverbal communication because they can reveal deeper human emotions as well as intent. The identification of facial expressions during social communication is a difficult process (Song et al. 2018). Deep Learning has proven to be superior to image processing methods for facial emotion detection. The six fundamental emotions that have been characterised historically include joy, sorrow, surprise, and fear. A FER (Facial Emotion Recognition) is a method that attempts to understand multiple facial muscles and classify them into the basic emotions mentioned above (Song et al. 2018).

This project aims to create a machine learning system that can be used in practice. Face expressions can be used to identify emotions and process them. It discusses three steps that are involved in emotion detection: facial, heart, and hand detection, features extraction and emotion classification (Kumar et al. 2020).

14.1.2 Visual Sentiment Analysis

Facial emotions are important in nonverbal communication because they can reveal deeper human emotions as well as intent. It is a challenging task to identify facial emotions in social communication. Deep Learning has proven to be superior to image processing methods for facial emotion detection. Happiness, sadness, surprise, and fear are the six basic emotions that have been described historically.

A FER (Facial Emotion Recognition) is a method that attempts to understand multiple facial muscles and classify them into the basic emotions mentioned above. FER is often implemented using many hand-crafted features, such as SIFT, LBP, and HoG. However, this approach is not able to simultaneously address multiple factors. Convolutional neural networks (CNN) have been shown to be promising when applied in FER research.

This project aims to create a machine learning system that can be used in practice. Face expressions can be used to identify emotions and process them.

Fig. 14.2 Picture showing women with emotion (Ding et al. 2017)

Convolutional networks (ConvNets) have been applied to a number of issues, including high-dimensional shallow feature encodings, visual identification problems, and large-scale picture classification systems. In a number of image classification challenges, CNN-based algorithms have been shown to attain the best levels of accuracy (Fig. 14.2).

14.1.3 Deep Learning

It is a sub-field of machine learning that depends only on ANNs (Artifi-cial Neural Networks). The human brain is portrayed as being mimicked by neural networks, and the same is true of deep learning. Deep learning has the advantage that not everything needs to be explicitly programmed. In deep learning, we must train a model on a dataset and then refine it until it makes nearly accurate predictions on both the testing and validation datasets (Sharma and Sharma 2021). Deep learning models are very helpful in resolving the dimensionality issue since they can focus on specific features on their own, with barely any input from the programmer. The DL model can be broadly divided into two components (Fig. 14.3).

Feature Extraction Phase

In this phase, authors train deep architectures on a large dataset by extracting a feature using the cascade of different layers. Authors simply input the images and then feed them to other layers.

Classification

At this stage, the photos are categorised into the relevant class. In Fig. 14.4, the relationships between AI, ML, and DL are mentioned, as seen in the diagram below.

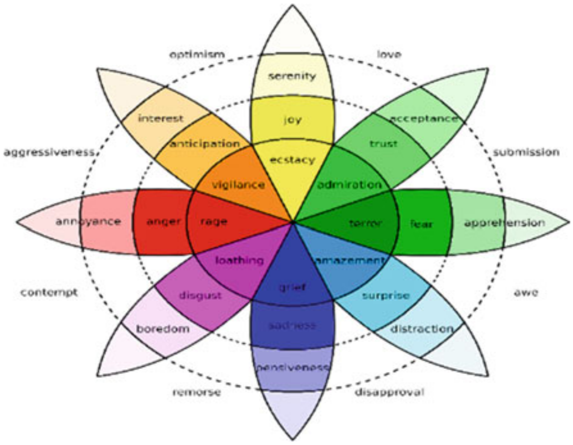


Fig. 14.3 Plutchik's wheel of emotions

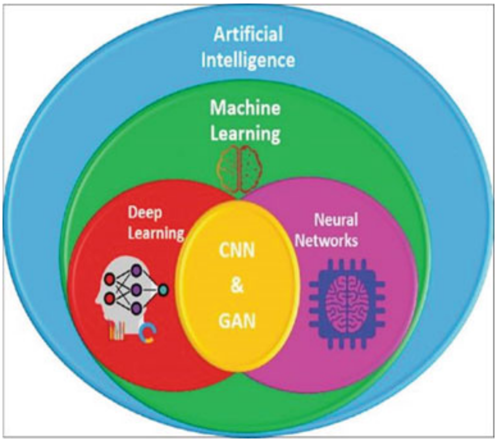


Fig. 14.4 Relationship between neural network, DL and CNN (Meng et al. 2017)

14.1.4 Convolutional Neural Networks

CNN is a Deep Learning calculation that can detect symbolism, assign esteem (intelligible and prejudicial instruments) to the image's various components/components, and distinguish one from the other. In comparison to previous stage calculations, ConvNet requires more preparation. While the old channels are handmade, with s, ConvNets can pre-use these channels' highlights with sufficient practice neurons that react to upgrades just in the limited area of the review field, also called as the Reception Field. The assortment of such fields penetrates to cover the whole obvious surface. CNN, which can, without much of a stretch, recognize and arrange objects with negligible preparation, prevails concerning breaking down visual pictures. It can, without much of a stretch, recognize the necessary highlights by its numerous lines structures. CNNs' indeed are the basic machine learning algorithms, for example, where a more powerful model increases artificial intelligence by demonstrating various sorts of human biological brain activity. A deep neural network has an input layer, various hidden layers, including completely connected layers, normative layers, multiple convolution layers, average pooling, and layers with multiple connections and an output layer. Few of these stripes are convolutional, which means they utilize a mathematical figure to pass information from one layer to the latter. This is same as some of the functions of the human visual brain.

The general CNN model is represented below in Fig. 14.5.

This stage involves moving each filter to every location on the image that is feasible. In a similar manner, drag the filter around the image to check how the feature lines up. Finally, for one feature, we will get the output shown in Fig. 14.6. After repeating the process for the other two filters, we obtain the convolution results for all of them.

Activation Layer

The actual work in convolutional neural networks provides a connection between the model's various layers. This layer contains a few functions such as ReLU, sigmoid, and SoftMax.

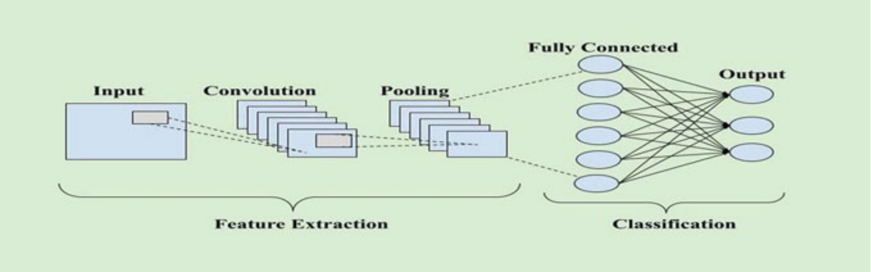
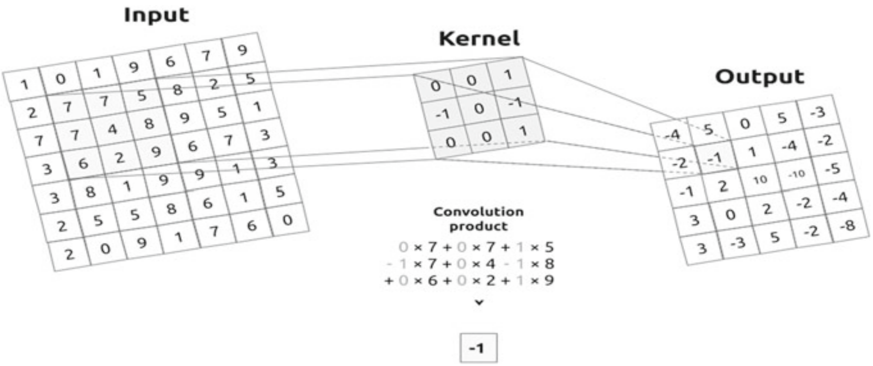


Fig. 14.5 General CNN model



1	0	1	9	6	7	9
2	7	7	5	8	2	5
7	7	4	8	9	5	1
3	6	2	9	6	7	3
3	8	1	9	9	1	3
2	5	5	8	6	1	5
2	0	9	1	7	6	0

0	0	1
-1	0	-1
0	0	1

-4	5	0	5	-3
-2	-1	1	-4	-2
-1	2	10	-10	-5
3	0	2	-2	-4
3	-3	5	-2	-8

$$0 \times 7 + 0 \times 7 + 1 \times 5 - 1 \times 7 + 0 \times 4 - 1 \times 8 + 0 \times 6 + 0 \times 2 + 1 \times 9$$

-1

Fig. 14.6 Convolution (Meng et al. 2017)

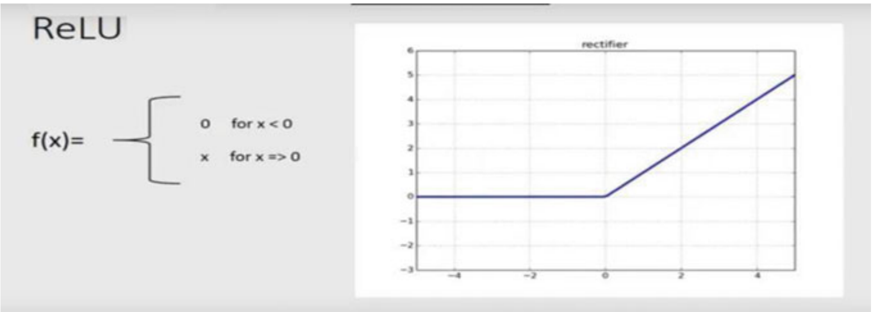


Fig. 14.7 ReLU function (Liu et al. 2017)

ReLU Layer

A node is activated by the Rectified Linear Unit (ReLU), a function that is used to determine when an input has crossed a predetermined threshold. As shown in Fig. 14.7, the output has a linear connection with the dependent variable when it reaches a particular threshold but is zero when the input is less than zero. All negative values in the convolutional results are changed to zeros in this layer. To stop the values from adding up to 0, this is done.

Dropout Function

In CNN, the dropout function is crucial. It helps prevent the over-fitting issue. Overfitting is a sign that a model has gotten too dependent on a set of data and will no longer work well with different sets of data.

Flatten Function

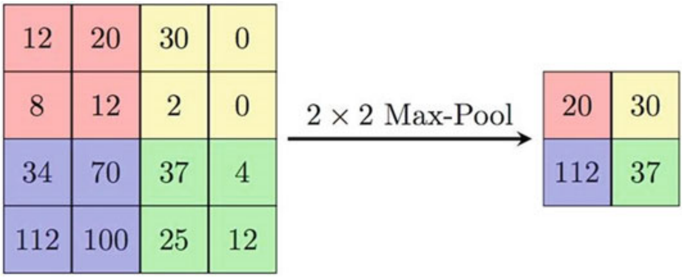
Using this function, the pooled feature map is transformed into a single-dimensional array and then forwarded to the following layer.

SoftMax

SoftMax builds on this idea by developing an universe with multiple classes. SoftMax gives each class in a multi-class problem a decimal probability. These decimal probabilities must add up to 1.0. This extra restriction makes training converge more quickly than it would otherwise.

Pooling Layer

This layer reduces the image to a smaller frame. In this instance, the max-pooling function is utilised. Following are the steps: Choose a stride, a window size (often 2 or 3), walk the matrix over the ReLU results, and choose the highest value from each window. Let’s examine the outcomes of pooling the first filtered image with either a matrix size of 2 or a stride of 2. The largest or maximum value is 1, so we keep track of it and advance the matrix by two steps. The pooling of a single 2*2 matrix



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

Fig. 14.8 Pooling layer (Yang et al. 2018)

is shown in the figure. After pooling the entire image, we obtain the result shown in Fig. 14.8.

Batch Normalization

A deep neural network training technique called batch normalisation normalises the two. The final layer 2*2 matrix 2 input to a layer for each mini batch is this pooling. This approach will decrease the number of epochs required to train the model.

Fully Connected Layer

In essence, the convolutional network's fully connected layer learns a (possibly non-linear) function in the pertinent low-dimensional, somewhat invariant feature space. The classification occurs in this layer, which is the bottom layer. Our filtered and condensed photos are combined into a single list or vector in this step.

14.1.5 Transfer Learning

Transfer Learning (TL) is an machine Learning optimization technique that emphasizes transferring knowledge gathered while solving one task to another but a similar task (Priyavrat and Sikka 2021). Using transfer learning, the training cost is minimized, accuracy is improved, and low generalization error is obtained.

To obtain the best performance in comparison to existing CNN models in different implementations, building and training a CNN architecture from scratch is an arduous procedure. Therefore, different models may be retrained according to the applications and is also used for feature engineering. Machine Learning and Knowledge Discovery in Data (KDD) have made huge colossalress in numerous knowledge engineering areas, including classification, regression, and clustering. The pre-trained model or wanted model segment can be incorporated straightforwardly into another CNN model (Aggarwal et al. 2023). The weight of the pre-trained models might be freezing in some applications; during the development of the new models, and weight can be updated with slower learning rates, which allows the pre-trained model to behave like weight initialization when the new model is learned. The pre-trained model may also be utilized as a weight initialization, classifier, and extractor. Firstly, we use the transfer learning approach, but it doesn't provide better results. After that we fine-tuned hyper parameters of our DL models and subsequently fine tuning provides us better results.

14.1.6 Fine Tuned CNN Models

Building a deep learning model from scratch is no easy task. Here some changes in the architecture of the deep learning model can be done as per the problem to be solved. Fine-tuning a deep learning model is a required step to improve the precision of anticipated outcomes (Pall et al. 2022). We gradually update the weights beginning with the most minimal level layers and working our way to the top. It learns a lot from pre-trained weights while training fine-tuned models. In the wake of training and testing, we can contrast our networks.

14.1.7 Traditional Methods

Different machine learning algorithms were employed in conventional approaches for picture identification and categorization (Pall et al. 2022).

Workflow of Traditional Methods

- Pre-Processing: This step is mainly done to remove noise from the image. This step mainly resizes the images. Various filters can be used in this step. This step can also help in increasing the accuracy.
- Object segmentation: This step mainly focuses on decreasing the search time by segmenting the image. This step helps in the detection of an object or region of interest.
- Feature Extraction: This step helps in the extraction of the desired features. Various machine learning can be used in the identification of desired features.
- Classification: This step helps in classification i.e. this step determines to which category the determined object belongs.

In machine learning a huge dataset can't be processed. At one value in the machine learning the model stops giving better accuracy and remains confined to one.

14.1.8 Research Objectives

- Development of various models coupled with Transfer Learning which will help in the Visual Sentiment Analysis.
- To use Facial Emotion images for the detection and prediction Various Emotion.
- To increase the accuracy of various pre-trained models in different datasets.
- To give a comparative analysis of different models in the detection process.
- The presented pre-trained DL models have an end-to-end structure that is totally autonomous and does not require custom feature extraction techniques.

14.1.9 Thesis Organization

The work has been organized into six chapters: Sect. 14.1 gives a brief introduction along with motivation and objectives. Section 14.2 discusses literature surveys of the work. Section 14.3 gives the problem definition and model architecture. Section 14.4 describes Dataset and Evaluation metrices and discusses the experimentation and results. Section 14.5 discusses the conclusion and future scope.

14.2 Literature Survey

An overview on visual sentiment analysis is represented in this section. VSA is a very recent area various sections are defined to show the previous work. At the end of the Literature review there are summary of all the related work.

Ortis et al. (2021) work used FER-2013 as a origin of design new CNN models. They developed these two CNN architectures. Because they were simple and used fewer training parameters, the architectures created expressly for the FER-2013 dataset were adequate. Both models are proposed on the FER-2013 dataset, their accuracy was better than 65%. Their models were able to predict the model and had the lowest number of parameters to train than other models, but were still capable of performing give human-like accuracy (65%). It was the best model for the job dataset.

Studies in the subject of visual sentiment detection are mainly concerned with issues like modelling, detecting, and utilising emotions communicated through facial or physical gestures (Kaya et al. 2012).

One new area of research is facial expression and nonverbal sentiment analysis. Researchers have used multimodal sentiment analysis on videos; however, more work needs to be done in relation to visual sentiment.

Researchers investigated the sentiment of adjectives over 100 images annotated by 42 subjects. They found it was possible to predict whether a pair of adjective–adjective words were present in an image by using aspects of light, saturation and sharpness to improve sentiment prediction.

Gonçalves et al. (2013) introduced an AI that used machine learning and linear SVM outputs to create visual sentiment analysis. ANPs, a semantic construction used in the SentiBank approach, paired “adjectives” and “nouns” for visual detectability, producing pairs like “cute bear,” “beautiful sunrise,” “tasty meal,” and “dreadful accident.”

The ANP model, proposed in Hasan et al. (2018) for constructing a visual sentiment ontology, inspired the SentiBank detector bank. The process involved the use of Flickr and YouTube APIs to identify images with emotive qualities. By part-of-speech tagging and extracting candidate pairs based on sentiment strength, named entities, and popularity, individuals were then able to produce a pool of adjective-noun pair candidates to search.

In the paper (Al-Halah et al. 2019) they have presented “Sentiment Networks with Visual Attention (SentiNet-A) is an layered network that investigates visual attention in order to improve picture sentiment analysis. A multi-layer neural network has been developed and implemented into a CNN-based image recognition system. Extensive trials on two benchmarks back up our assertion and proposition” (Corchs et al. 2019).

In the paper (Vadicamo et al. 2017) they have presented a hybrid approach for real-time sentiment analysis using a combination of text and picture modalities was proposed in research (info-graphic). For text, picture, and mixed data, the model exceeded the baselines, with sentiment classification accuracy of around 88, 76, and 91%, respectively (Gonçalves et al. 2013). The decision module might also aid in the detection of neutral ambiguities that were indicative of sarcasm.

The issue of visual sentiment based on convolutional neural networks is addressed in this study, where the sentiments are predicted utilising a variety of affective cues (Hasan et al. 2018). They have described a two-branch, end-to-end, weakly supervised deep architecture for learning discriminative representations.

This study (Al-Halah et al. 2019) proposes a novel cross-modal technique for semantic content correlation that links captions to images. An image-text combination is identified by this model using a joint attention network, and the relationship between the words in the caption is then determined using that information as the query (Al-Halah et al. 2019). The prediction of picture sentiment is based on the correlation of two modalities, textual meaning and visual content.

In this paper (Corchs et al. 2019), MASAD has 38k samples with image-text pairs and 57 aspects with seven domains. Extensive tests on the MASAD dataset have revealed that models trained on this dataset have a strong generality potential. The following research directions, we feel, are worth investigating: More expressive model architectures for learning inter multimodal knowledge are being developed. We will try to figure out how to extract the context and forecast the sentiment at the same time.

Recently, researchers have been creating machine learning models specific to sentiment analysis for visual media, such as photographs or videos. They were able to achieve higher accuracies of above 20% on sentences involving adjectives and nouns (Vadicamo et al. 2017). In contrast to prior models that employed convolutional neural networks for image classification and only classifiers for sentiment analysis with SVMs, others constructed a model on ImageNet before training it particularly

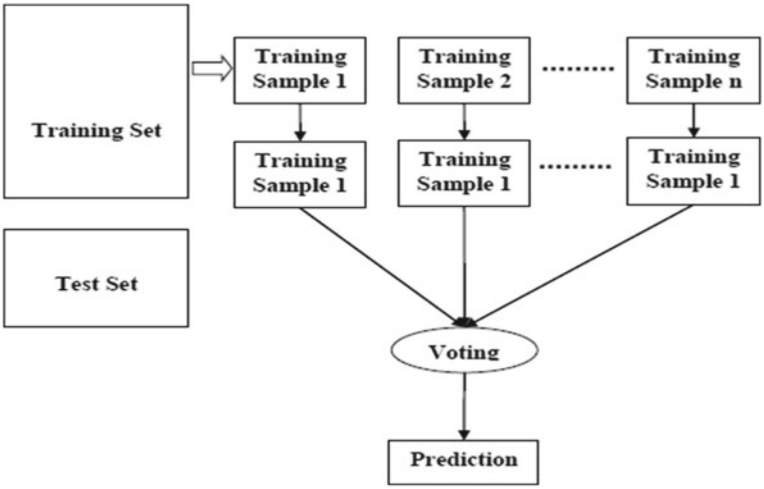


Fig. 14.9 General training set categorization

for sentiment detection with SVMs (Machajdik and Hanbury 2010). The researchers also considered the inclusion of LSTM (long short-term memory) in order to create more naturalistic sentence summaries by incorporating both object grounded nouns and affective descriptors.

Visual sentiment analysis can increase accuracy, covering a wider variety of languages and media types. It is not always accurate, missing the context of a situation or an entity (Fig. 14.9).

14.3 Methodology and Implementation

CNNs will be used for this project. The dataset will then be selected. Then, we will do the appropriate pre-processing. Then we'll be Divide the data into validation, testing, and training. Next, we create the model from the data and train it is using the dataset. This will produce the desired result. This proposed method aims to predict 6 facial expressions viz. You can do this by using CNN's Sequential Model. This model will be further enhanced with the layers of different algorithms that can be used. These algorithms will be used for activation functions, learning rates etc.

14.3.1 Sentiment Classification

Various machine learning approaches that can be used for this purpose are discussed below in detail.

CNN

CNN is a DL approach which is used to make the machines learn from their past experiences. This is a type of supervised technique in which data is first fed to the model and then it predicts the unseen corpus.

Support Vector Machine

Support Vector Machine (SVM) is a machine-learning technique used for classification of sentiment in this study. Each sample is labelled corresponding to one of the classes from the given set of training samples. An SVM training model assigns new samples to each class. Multiclass SVM makes labels to instances which are assigned from finite set of elements.

K-Nearest Neighbor

KNN is a supervised ML approach. It follows a majority voting method for similarity to K-nearest neighbor which is calculated using any of the distance methods. Euclidean, Manhattan and Minkowski distance are the commonly used distance methods (Fig. 14.10).

14.3.2 Various Steps Involved in Visual Sentiment Analysis

Data Acquisition

In the data acquisition process, a trained data scientist finds datasets and Machine Learning models to train. Various methods are used to generate data from different sources are as follows:

- Data Generation
- Data Discovery
- Data Augmentation.

The dataset consists of a very huge number of pictures to get more accurate results. The data in this model consists of grayscale pictures of faces 48 × 48 pixels. 35,887 facial images of different emotions were collected to categorize into 7 different classes. These faces were captured using a smartphone and blurry images will not give proper results. Capturing the faces in the middle of the image was important to more reliably categorize their expressions." The test set has 3,589 cases, whereas the training set has 28,709 examples. And the validation set has the same 3,589 items as the test set.

Data Pre-processing

There are a few changes necessary to any image before it is run through the machine. You can change the number of pixels or shrink the size if not necessary and remove any additional noise. This will give you much more accurate results and higher accuracy. Additionally, data augmentation is carried out in this methodological stage. The

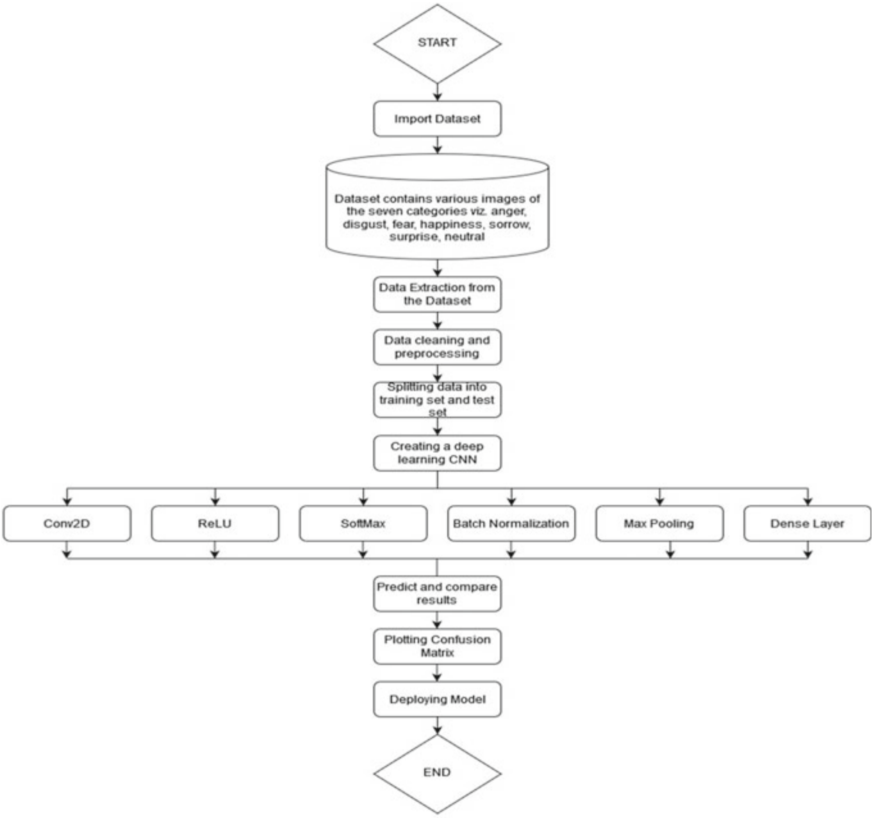

START
Import Dataset
Dataset contains various images of the seven categories viz. anger, disgust, fear, happiness, sorrow, surprise, neutral
Data Extraction from the Dataset
Data cleaning and preprocessing
Splitting data into training set and test set
Creating a deep learning CNN
Conv2D
ReLU
SoftMax
Batch Normalization
Max Pooling
Dense Layer
Predict and compare results
Plotting Confusion Matrix
Deploying Model
END

Fig. 14.10 Working of machine learning model

term "data augmentation" describes the process of greatly expanding the amount of photographs without actually taking new ones.

Feature Extraction

The various features that must be extracted for this project are:

Color-Histogram

The simplest visual feature taken from RGB colour channels is the Color Histogram. A 256-dimensional histogram displaying the distribution of pixel values may be obtained for each channel. The feature vector is then normalised to unit length and referred to as a feature vector (Fig. 14.11).

GIST Descriptor

For expressing real-world situations in a computer model, the GIST descriptor is proposed. The picture is initially filtered using numerous Gabor filters of varying sizes and orientations. Using the filters, you may capture the texture information

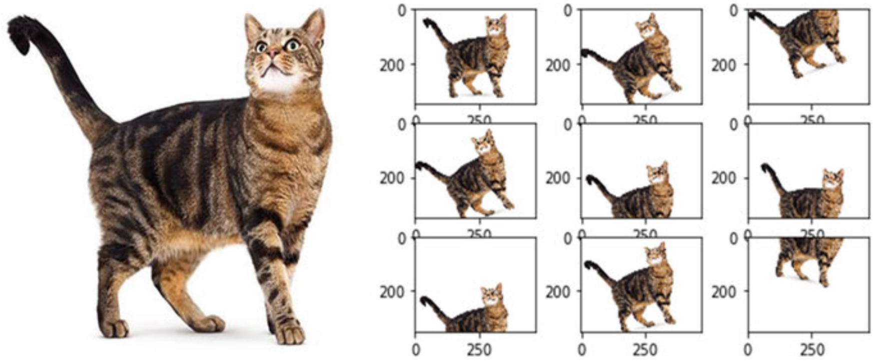


Fig. 14.11 Image augmentation

in the image. In this thesis, we investigate using three scales, with the number of orientations under each scale fixed at eight, eight, and four, respectively. We construct 20 feature maps in total, one for each of the 20 Gabor filters. We divided each map into four grids. Each grid's energy is defined as its average response value. All the energies are eventually merged into a 320-dimensional GIST feature vector.

Bag-of-Visual-Words (BoVW)

Bag-of-Visual-Words is similar to bag-of-words (BoW) in that the words are descriptors calculated on picture patches surrounding key points. The difference of Gaussian functions (DoG) applied in the scale space is used to locate the position of key points. The Hessian-Affine area detector is another popular approach for key points discovery. The dominant direction and size are determined by examining the gradient values around the identified key point. As a result, the extracted feature is insensitive to translation, scaling, rotation, and lighting changes. Instead of extracting local features around key points, it has been discovered that the approach employing intensively sampled local areas may obtain equivalent results for visual identification.

14.3.3 Building and Designing the Model

The various methods used for classification of this thesis are CNN, SVM and KNN. So, the working of all three of them is discussed below.

Convolutional Neural Networks

CNN is a deep-learning algorithm which takes an input image, assigns various weights and biases to the image so that it can be differentiable from one another. It mainly consists of four layers. The explanation of the various layers is given below (Fig. 14.12).

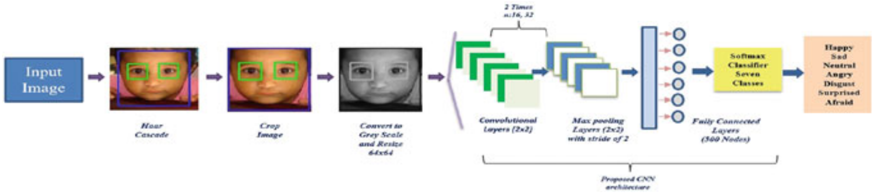

Input Image
Haar Cascade
Crop Image
Convert to Grey Scale and Resize 64x64
2 Times n:16, 32
Convolutional Layers (2x2)
Max pooling Layers (2x2) with stride of 2
Fully Connected Layers (500 Nodes)
Softmax Classifier Seven Classes
Happy
Sad
Neutral
Angry
Disgust
Surprised
Afraid
Proposed CNN architecture

Fig. 14.12 CNN model overview (Hamester et al. 2015)

Convolution Layer

This layer of convolution neural network is used to perform mathematical functions in the model.

Pooling Layer

To summarize the images in the input, convolutional layers apply learned filters to the image. The dimensions of the filter being applied to the image is smaller than the size of the pixels setting up the image.

Activation Layer

In CNN, the actual work gives a connection b/w the different layers of the model. A few functions like ReLU and sigmoid are present in this layer.

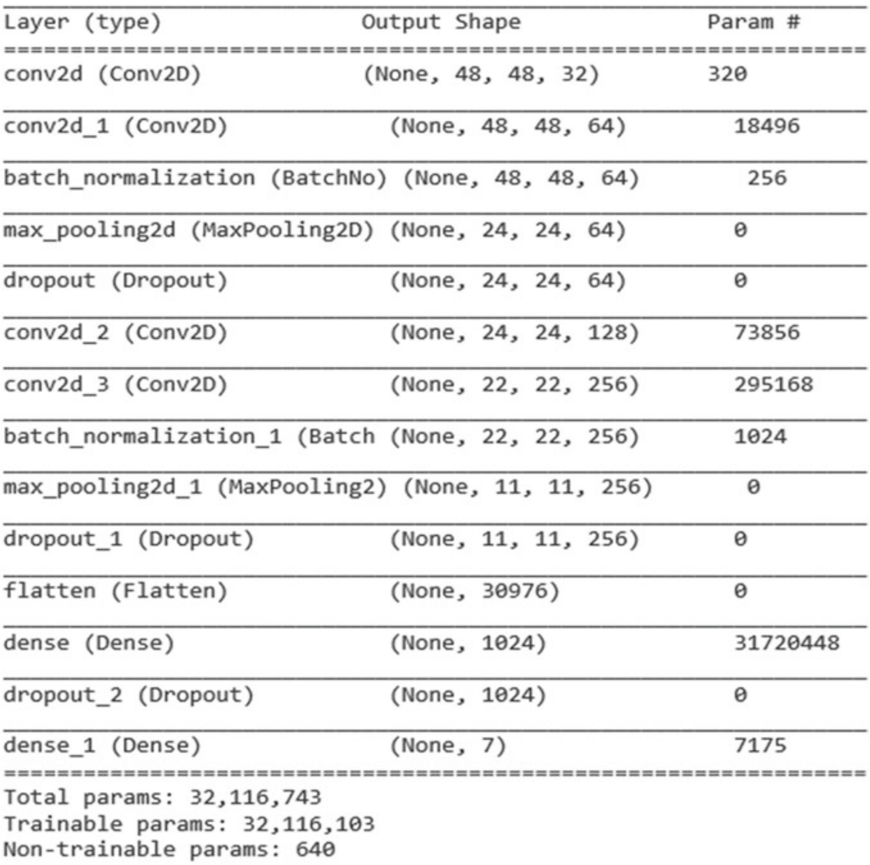
Fully Connected Layer

A fully Connected layer is a feed-forward NN. It's the last few layers in the network and creates output for the entire input. Recognition and classification on the image are performed in this layer.

The CNN model was chosen for this task because it requires less human oversight and pre-processing of the data. It is a self-learning algorithm that identifies key components in images that have been divided into test and train set in an 20:80 ratio.

With this model, the picture size is initially set at 48*48. The picture is then sent into the first convolution layer. The picture is then sent through a convolutional with 64 kernels of 15*15 size. The process is repeated up until the input size reaches 1*11. The input images are processed in this layer before the mathematical operations are carried out.

The max-pooling layer is an additional crucial layer. The max-pooling layer's primary goal is to reduce the size of the feature map. The main information source for this layer is the feature map. The Dense Layer is the most interconnected layer in the model, concluding that every neuron from earlier layers is connected to the layer below it. The output data is returned by this layer. The proposed model's model summary is shown below.

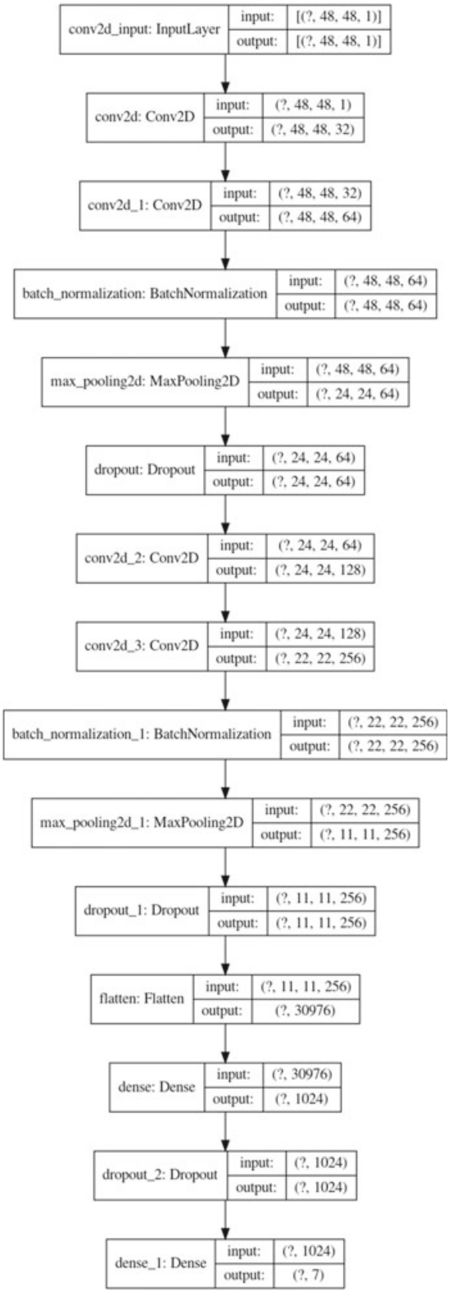


```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 32)        320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 48, 48, 64)        18496
_________________________________________________________________
batch_normalization (BatchNo) (None, 48, 48, 64)        256
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 24, 24, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 24, 24, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 128)       73856
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 22, 22, 256)       295168
_________________________________________________________________
batch_normalization_1 (Batch (None, 22, 22, 256)       1024
_________________________________________________________________
max_pooling2d_1 (MaxPooling2) (None, 11, 11, 256)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 11, 11, 256)       0
_________________________________________________________________
flatten (Flatten)            (None, 30976)             0
_________________________________________________________________
dense (Dense)                (None, 1024)              31720448
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 7175
=================================================================
Total params: 32,116,743
Trainable params: 32,116,103
Non-trainable params: 640
```

Model: “sequential”

Support Vector Machine

Support Vector machine is a supervised machine-learning model implemented in this thesis. It is a very effective and accurate model so it is chosen to compare with CNN. SVM divides the linear and non-linear data into classes. In SVM, we need to maximize the margin between two classes. The classification that helps finding the maximum optimal margin is called as hyper-plane (Fig. 14.13).


conv2d_input: InputLayer
input: [(?, 48, 48, 1)]
output: [(?, 48, 48, 1)]
conv2d: Conv2D
input: (?, 48, 48, 1)
output: (?, 48, 48, 32)
conv2d_1: Conv2D
input: (?, 48, 48, 32)
output: (?, 48, 48, 64)
batch_normalization: BatchNormalization
input: (?, 48, 48, 64)
output: (?, 48, 48, 64)
max_pooling2d: MaxPooling2D
input: (?, 48, 48, 64)
output: (?, 24, 24, 64)
dropout: Dropout
input: (?, 24, 24, 64)
output: (?, 24, 24, 64)
conv2d_2: Conv2D
input: (?, 24, 24, 64)
output: (?, 24, 24, 128)
conv2d_3: Conv2D
input: (?, 24, 24, 128)
output: (?, 22, 22, 256)
batch_normalization_1: BatchNormalization
input: (?, 22, 22, 256)
output: (?, 22, 22, 256)
max_pooling2d_1: MaxPooling2D
input: (?, 22, 22, 256)
output: (?, 11, 11, 256)
dropout_1: Dropout
input: (?, 11, 11, 256)
output: (?, 11, 11, 256)
flatten: Flatten
input: (?, 11, 11, 256)
output: (?, 30976)
dense: Dense
input: (?, 30976)
output: (?, 1024)
dropout_2: Dropout
input: (?, 1024)
output: (?, 1024)
dense_1: Dense
input: (?, 1024)
output: (?, 7)

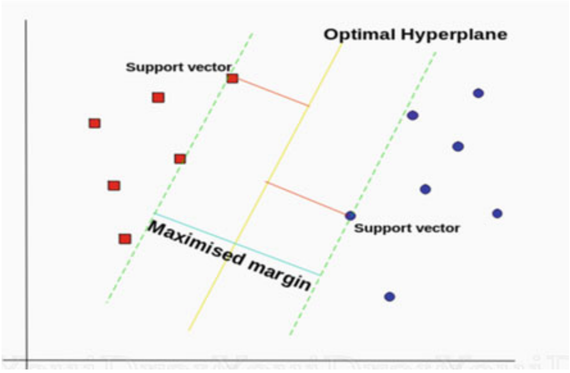


Fig. 14.13 SVM hyperplane

14.3.4 Model Flow

SVM Kernels are used to add more dimensions to the low dimensional space to make it easier to divide the data. So, the various kernels that can be used in SVM are as follows.

Linear Kernel

Linear kernel is the type of kernel which is used when the data can be separated linearly which means using a single line. It is one of the most commonly used kernels of the SVM as it can train the large data models.

Polynomial Kernel

This kernel function represents the similarity of vectors in the set of data in a feature space over the polynomials.

Radial Basis Function Kernel

This kernel is used for performing various kernelized learning algorithms. It is generally used for classification purpose.

In this project, the data is first divided for SVM classification into various classes depending upon the corresponding diseases in plants. Then, the linear function kernel is used as it is a fast learning algorithm and is helpful for large data-sets.

The size of the iterations in our model is kept as 500 which means the data is passed through the model 500 times to train the model (Fig. 14.14).

K-Nearest Neighbor

KNN is an algorithm that memorizes all the available classes and detects the new data on that basis. K is the number of nearest neighbors in KNN. We have to define the value of K. For e.g. If we take the value of K as 3 that means the 3 points from the object will be selected and then their distance will be evaluated. The point with the closest distance will be chosen. The distance measures in choosing the K-nearest can be any of the following. Some most commonly used distances are explained below.

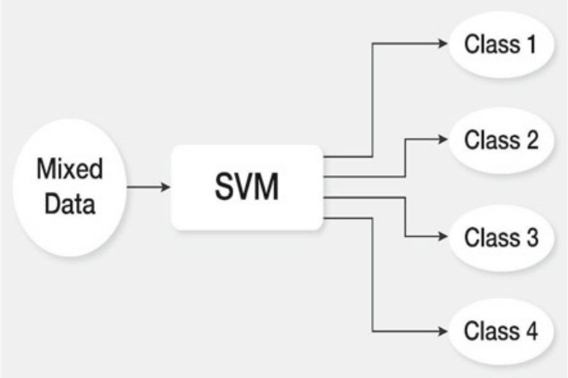


Fig. 14.14 Classification of data in SVM (Sun et al. 2016)

- Euclidean Distance

This distance is the most commonly used method in KNN. It is the measure of a line between 2 points (×1 and ×2) on a straight line.

- Manhattan Distance

This distance is the sum of difference between the x-coordinate and y-coordinate (Fig. 14.15).

$$d = |x1 - x2| + |y1 - y2|$$

Prediction is slow in the KNN algorithm and is called as a lazy learner because it does not have a classifier in the learning phase. This algorithm simply memorizes the whole dataset and there is no training in the KNN algorithm. Each time the item that is to be predicted is given to the KNN model, it searches for its nearest

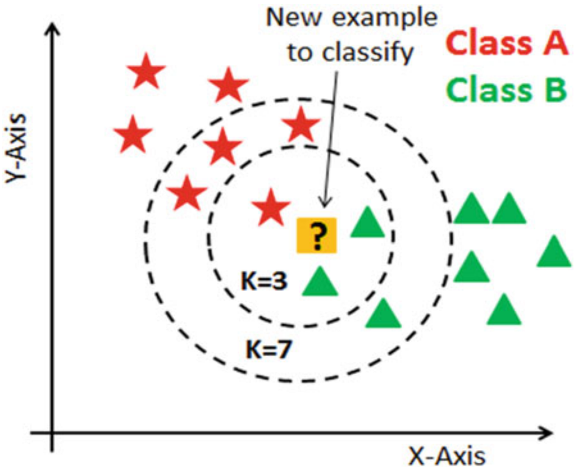


Fig. 14.15 KNN model overview (Liu et al. 2019)

neighbors in the whole dataset. So, the testing process in this model is quite lengthy and time-consuming.

14.4 Dataset and Evaluation Metrics

14.4.1 Facial Emotion Recognition (FER) Dataset

'Grayscale images of faces' that were automatically registered make up the information. It can be difficult to categorise each face based on the emotion that is shown in their expression. "emotion" and "pixels" are the two columns in "train.csv." The "feeling" column contains a number code that spans the entire range of emotions visible in the image, from 0 to 6, inclusive. A string of values encased in "quotes for each image" serves as the "pixels" column's description of row-major pixel values. Based on the pixel values in the "train.csv," the model's objective is to identify the emotion that is present (Fig. 14.16).

Dataset Details

Sr. no.	Class name	# Training image	# Validation image	# Testing image
1	Angry	3995	958	958
2	Disgust	1987	497	497
3	Fear	1760	440	440
4	Happy	2008	502	502
5	Neutral	1816	454	454
6	Sad	1826	456	456
7	Surprise	1683	421	421

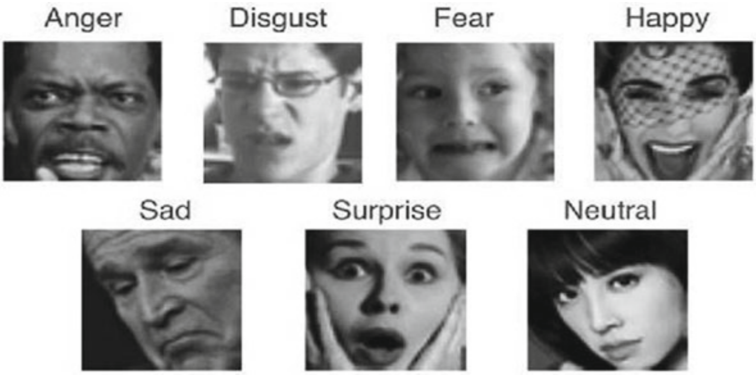


Fig. 14.16 Sample image from FER dataset

14.4.2 Evaluation Metrics

With the aim of analyzing the performances of DL models on the basis of various hyper parameters like Validation Accuracy, precision, accuracy, recall and F1 score. Confusion Matrix, and AUC curve have been calculated in this study. In this chapter, definitions of various parameters used in this study are given and then in the next chapter these parameters have been drawn out for different models.

- **Accuracy**

Use Evaluation Metrics to make the right Decision and run Agile. It is the outcome of the true positives and true negatives., It is also defined as performance measure for model. It can be calculated as

$$\mathrm{Acc} = [(\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{FN} + \mathrm{TN} + \mathrm{FP})] * 100\%$$

- **Training and Testing Accuracy**

The term “training accuracy” refers to the utilization of identical pictures for both training and testing. The test accuracy denotes the trained model’s ability to recognize images that were not utilized during training.

- **Validation Accuracy**

It is the type of accuracy which is measured on the sample of data held back while training the model. This kind of accuracy gives an estimate of the model skill.

- **Precision**

It is defined as the fraction of the TP labelled by sum of the true positive and false positives.

$$\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$$

- **Recall**

The recall is defined as fraction of the sum of positive by the number of positive accurately classified as Positive. It is the capability of the model to detect the correct positive and is defined as

$$\mathrm{Recall} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$$

- **F1-Score**

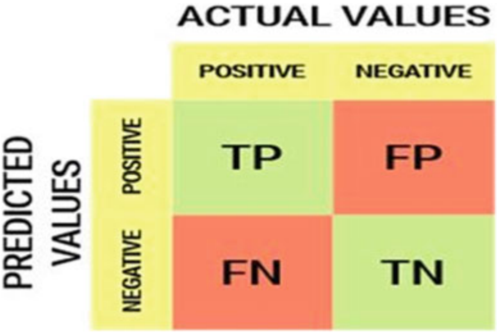
It is known as the “harmonic mean” of the model’s recall and accuracy. In other words, it is the weighted average of Precision and Recall. Typically, increasing accuracy causes recollection to decline, and vice versa. The authors occasionally want to consider both precision and recall at once. F1 score can be calculated as

$$\text{F1-Score} = 2 * (\text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall})$$

The authors can compute the F1 Score for each class in a multi-class classification since the authors know the Precision and Recall for each class.

14.4.3 Confusion Matrix

A confusion matrix is used to gauge how well the classifier or model is doing. An error matrix is a list of all correct and incorrect predictions. Both the actual values and the values that the classifier had predicted are disclosed. Its main function is to rate the effectiveness of classification in machine learning systems, especially supervised learning algorithms. As shown below, the authors would have a 2 × 2 matrix with four values for a binary classification problem. Below is an illustration of a confusion matrix.



		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

True Positive (TP)

- When the predicted and observed values agree.
- When the value matched what the model predicted and was favourable.

True Negative (TN)

- When the actual value and model predictive was negative and same.

False Positive (FP)—Type 1 error

- When the model create a positive result but the actual value was negative. Also referred to as the Type 1 mistake.

False Negative (FN)—Type 2 error

- When the model predicted a negative value but the actual value was positive. Also known as the Type 2 error.

14.5 Experimentation and Results

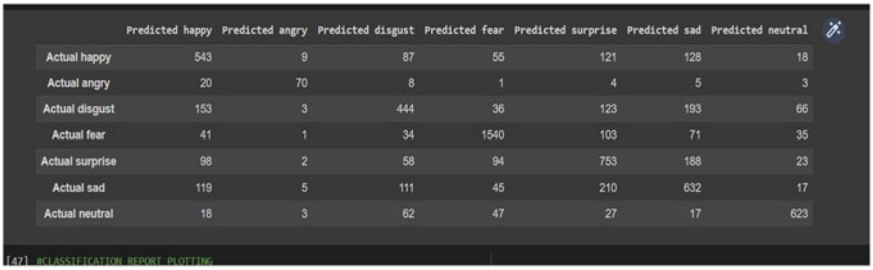
The authors now begin training our model with the FER-2013 dataset. As discussed earlier, the training set would consist of 28,821 images. There are already labels provided so that tells the model the kind of expression each image input has. For this example, the authors make it go through with 100 epochs as the authors think this would be more convenient. While the training is processing, the authors can observe that the accuracy of the model keeps on increasing. The final accuracy that the authors achieve by training the model is ≈89%. The authors can now go ahead and use the test set.

Method	Accuracy rate (%)
GoogleNet	65.20
(VGG) + (SVM)	66.31
Convolution + inception layer	66.40
BW (bags of words)	67.40
Attentional ConvNet	70.02
ARM (ResNet-18)	71.38
ResNet	72.40
VGG	72.70
CNN (this work)	89.23

We now begin testing our model with the FER-2013 test set. Our test set consists of 7067 images. For testing the syntax is model.predict (test_set). After running the code, we compare the output with the test labels. With this comparison we can tell how accurate our model turned out to be. The resulting accuracy we got after testing our model came out to be ≈90%

The results of plotting the confusion matrix and classification report are given in the table below. How the confusion matrix looks like can be roughly seen. The predicted labels outline the results of the model's predictions of the expressions while the actual labels outline the actual expression of the image. There is also a heatmap that depicts the difference in the range of accuracies of each predicted label and the actual label (Fig. 14.17).

Classification report of all the scores is given below. This points out the precision of the various labels and shows the recall, F-1 score and support (Figs. 14.18 and 14.19).



	Predicted happy	Predicted angry	Predicted disgust	Predicted fear	Predicted surprise	Predicted sad	Predicted neutral
Actual happy	543	9	87	55	121	128	18
Actual angry	20	70	8	1	4	5	3
Actual disgust	153	3	444	36	123	193	66
Actual fear	41	1	34	1540	103	71	35
Actual surprise	98	2	58	94	753	188	23
Actual sad	119	5	111	45	210	632	17
Actual neutral	18	3	62	47	27	17	623

```
[47] #CLASSIFICATION REPORT PLOTTING
```

Fig. 14.17 The output snapshot of the confusion matrix table

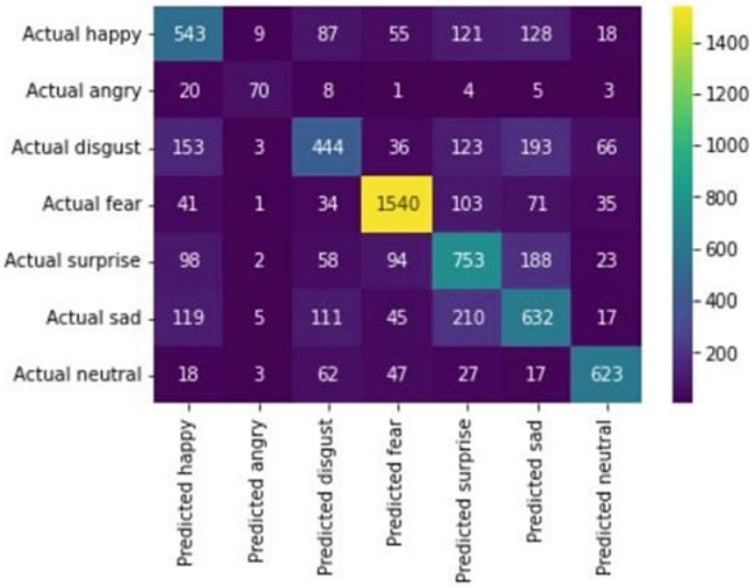
	Predicted happy	Predicted angry	Predicted disgust	Predicted fear	Predicted surprise	Predicted sad	Predicted neutral
Actual happy	543	9	87	55	121	128	18
Actual angry	20	70	8	1	4	5	3
Actual disgust	153	3	444	36	123	193	66
Actual fear	41	1	34	1540	103	71	35
Actual surprise	98	2	58	94	753	188	23
Actual sad	119	5	111	45	210	632	17
Actual neutral	18	3	62	47	27	17	623

Fig. 14.18 Confusion matrix

Classification Report

	Precision	Recall	F1-score	Support
Happy	0.49	0.63	0.55	961
Angry	0.70	0.60	0.65	111
Disgust	0.60	0.39	0.47	1018
Fear	0.84	0.84	0.84	1825
Surprise	0.55	0.63	0.59	1216
Sad	0.51	0.47	0.49	1139
Neutral	0.79	0.77	0.78	797

Based on the results which we got in our research it indicates that application deep learning can be useful to predict Facial emotions. Our system can be utilized in many ways to benefit efficiency. In the future, we may investigate additional deep learning



	Predicted happy	Predicted angry	Predicted disgust	Predicted fear	Predicted surprise	Predicted sad	Predicted neutral
Actual happy	543	9	87	55	121	128	18
Actual angry	20	70	8	1	4	5	3
Actual disgust	153	3	444	36	123	193	66
Actual fear	41	1	34	1540	103	71	35
Actual surprise	98	2	58	94	753	188	23
Actual sad	119	5	111	45	210	632	17
Actual neutral	18	3	62	47	27	17	623

Fig. 14.19 Heatmap of confusion matrix

techniques to see more efficient results. This can be beneficial in developing AI that can read human facial expressions which will significantly help human–machine interaction.

14.6 Conclusion and Future Scope

In this study, the authors have done fine-tuning on deep learning models and proposed a small-sized CNN for the classification of Emotions in Facial Emotion Recognition dataset.

This work achieves state of the art accuracy of 90% and used the Dataset which consist of approx. 34 thousand of images with the proposed model which is based on Convolution. Different types of deep-learning models were used but the highest accuracy was achieved by CNN.

This proposed study has given better results than most Emotion Classification methods used in the literature survey. In future work, I would like to extend our work on different datasets and will try to improve the performance in the classification of Sentiment.

References

Aggarwal T, Sharma N, Aggarwal N (2023) Gunshot detection and classification using a convolution-GRU based approach. In: Noor A, Saroha K, Pricop E, Sen A, Trivedi G (eds) Proceedings of emerging trends and technologies on intelligent systems. Advances in intelligent systems and computing, vol 1414. Springer, Singapore. https://doi.org/10.1007/978-981-19-4182-5_8

Al-Halah Z, Aitken A, Shi W, Caballero J (2019) Smile, be happy:) emoji embedding for visual sentiment analysis. In: Proceedings of the IEEE/CVF international conference on computer vision workshops, pp 0–0

Corchs S, Fersini E, Gasparini F (2019) Ensemble learning on visual and textual data for social image emotion classification. Int J Mach Learn Cybern 10(8):2057–2070. Springer Science and Business Media LLC

Ding H, Zhou SK, Chellappa R (2017) Facenet2expnet: regularizing a deep face recognition net for expression recognition. In: 2017 12th IEEE international conference on automatic face and gesture recognition (FG 2017). IEEE, pp 118–126

Gonçalves P, Araújo M, Benevenuto F, Cha M (2013) Comparing and combining sentiment analysis methods. In: Proceedings of the first ACM conference on online social networks—COSN '13. ACM Press

Hamester D, Barros P, Wermter S (2015) Face expression recognition with a 2-channel convolutional neural network. In: 2015 International joint conference on neural networks (IJCNN). IEEE, pp 1–8

Hasan A, Moin S, Karim A, Shamshirband S (2018) Machine learning-based sentiment analysis for twitter accounts. Math Comput Appl 23(1):11. MDPI AG

Kaya M, Fidan G, Toroslu IH (2012) Sentiment analysis of Turkish political news. In: 2012 IEEE/WIC/ACM international conferences on web intelligence and intelligent agent technology. IEEE

Khorrami P, Paine T, Huang T (2015) Do deep neural networks learn facial action units when doing expression recognition? arXiv preprint arXiv:1510.02969v3

Kumar A, Srinivasan K, Cheng W-H, Zomaya AY (2020) Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data. Inf Process Manag 57(1):102141. Elsevier BV

Liu X, Kumar BV, Jia P, You J (2019) Hard negative generation for identity-disentangled facial expression recognition. Pattern Recognit 88:1–12

Liu X, Kumar B, You J, Jia P (2017) Adaptive deep metric learning for identity-aware facial expression recognition. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), pp 522–531

Machajdik J, Hanbury A (2010) Affective image classification using features inspired by psychology and art theory. In: Proceedings of the international conference on multimedia—MM '10. ACM Press

Meng Z, Liu P, Cai J, Han S, Tong Y (2017) Identity-aware convolutional neural network for facial expression recognition. In: 2017 12th IEEE international conference on automatic face and gesture recognition (FG 2017). IEEE, pp 558–565

Ortis A, Farinella GM, Battiato S (2020) Survey on visual sentiment analysis. IET Image Process 14(8):1440–1456. Institution of Engineering and Technology (IET)

Ortis A, Farinella GM, Torrisi G, Battiato S (2021) Exploiting objective text description of images for visual sentiment analysis. Multimed Tools Appl 80(15):22323–22346. Springer Science and Business Media LLC

Pall A, Sharma N, Sharma K, Wadhwa V (2022) A systematic review of deep learning techniques for semantic image segmentation: methods, future directions, and challenges. In: Handbook of research on machine learning

Priyavrat SN, Sikka G (2021) Multimodal sentiment analysis of social media data: a review. In: Singh PK, Singh Y, Kolekar MH, Kar AK, Chhabra JK, Sen A (eds) Recent innovations in computing. ICRIC 2020. Lecture notes in electrical engineering, vol 701. Springer, Singapore. https://doi.org/10.1007/978-981-15-8297-4_44

Sharma R, Sharma N (2021) Application of machine learning in precision agriculture. In: Mangla M, Satpathy S, Nayak B, Mohanty SN (eds) Integration of cloud computing with internet of things. https://doi.org/10.1002/9781119769323.ch8

Song K, Yao T, Ling Q, Mei T (2018) Boosting image sentiment analysis with visual attention. Neurocomputing 312:218–228. Elsevier BV

Sun M, Yang J, Wang K, Shen H (2016) Discovering affective regions in deep convolutional neural networks for visual sentiment prediction. In: 2016 IEEE international conference on multimedia and expo (ICME). IEEE

Vadicamo L, Carrara F, Cimino A, Cresci S, Dell'Orletta F, Falchi F, Tesconi M (2017) Cross-media learning for image sentiment analysis in the wild. In: Proceedings of the IEEE international conference on computer vision workshops, pp 308–317

Yang H, Ciftci U, Yin L (2018) Facial expression recognition by de-expression residue learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2168–2177