Emotion Recognition from visual Big-Data using Convolutional Neural Network (CNN)

Vinay Ramasare Kurmi - 10576078

Supervisor: Dr. Charles Ezenwa Nwankire

Masters of Science in Data Analytics
Dublin Business School

This dissertation is submitted for the degree of
Masters of Science in Data Analytics

January 2022

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Vinay Ramasare Kurmi - 10576078
January 2022

Acknowledgements

First and foremost, I would like to offer my sincere gratitude to my Dissertation Supervisor Dr. Charles Ezenwa Nwankire of MSc in Data Analytics course at the Dublin Business School. Dr. Charles Ezenwa Nwankire was always open to help me whenever I was in trouble or had questions about my research or writing. He consistently ensured that my Masters research paper was unique and my own work. He also corrected me at times when he felt any problems or issues in my research artefact or writing. He was always open for any type of discussions regarding the research. Every meeting with him was a great learning experience since he asked me to try and do investigation on a lot of options in the areas which were very new for me. He was always reachable by his mobile phone or e-mail. Also I would like to thank and I'm in-debt to other teaching faculty at Dublin Business School in order to help me learn and implement the necessary knowledge, technologies and tools in order to successfully accomplish my dissertation.

Lastly, my family deserves endless gratitude: my father for teaching me to appreciate history and storytelling, my mother for teaching me how to write with concise purpose, and my brother for teaching me that an assertion of dominance is not necessarily a bad thing. To my family, I give everything, including this.

Abstract

Human species has been using their facial expressions and facial muscles in order to effectively communicate for millions of years now. Expression and emotions have been captured and stored into many forms since then like paintings, pictures, videos etc. Research is also being conducted in various forms in order to successfully identify the emotion of a person with help of technologies and methods such as knowledge based techniques, statistical methods and hybrid approaches. Recently artificial intelligence has proven its effectiveness in classifying or predicting the human emotion based on accumulated data in form of audio, video, text etc... This research proposes to design a system to classify the real-time emotions from numerous facial expressions and its features while using techniques such as Convolution Neural Network(CNN) algorithm. Along with CNN, this research also proposes to utilize various other tools and technologies such as OpenCV library, Tensorflow, Keras, Python, Visual Studio Code, etc.

Keywords : Deep Learning, Convolutional Neural Network, Emotion Classification, Image Processing, Visual Sentiment Analysis, Machine Learning, Artificial Intelligence, FER-2013 Dataset.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Recognizing emotions is considered to be an chief potential for ones interpersonal skills and plays a vital role in communications. It can be very distinctive and complex task for a human being. We have been trying to recognize human emotions for decades now using our human instincts, however sometimes we fail to successfully identify the emotions and sentiments the opposite person is going through in the first instance. The aim of this research is to successfully recognize the emotions from visual data i.e. images using deep neural networks. The research has attempted to identify and overcome the deadlocks in current architecture and systems to identify human emotions. Human emotions can be primarily identified using human facial muscles and facial gestures resulting into expressing their emotions, feelings and opinion of others. The aim of this research is to increases the accuracy of success full prediction of the 7 human emotions namely happy, sad, angry, scared, surprise, neutral and disgust while utilizing the Facial Emotion Recognition dataset (FER 2013).

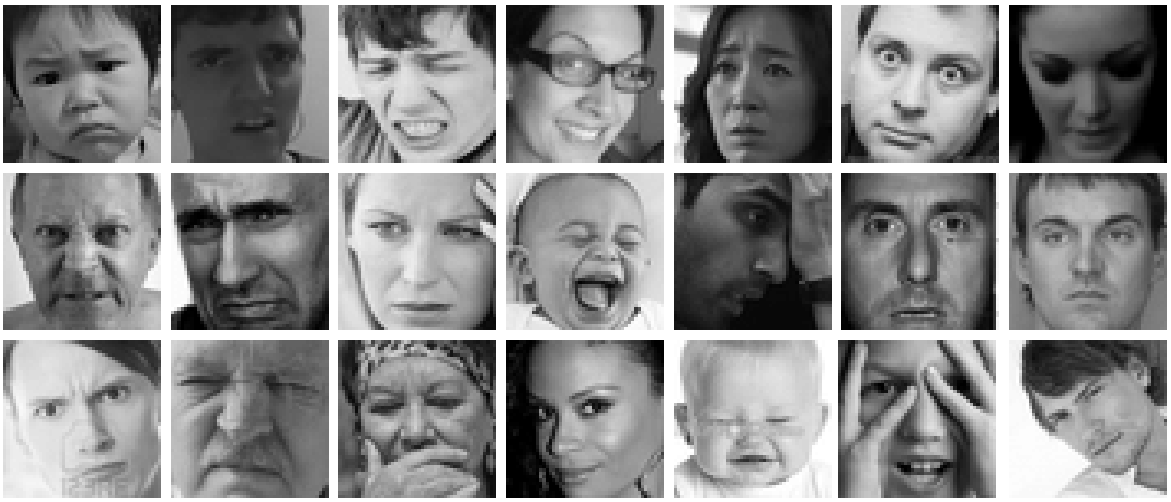


Fig. 1.1 Some sample images from FER 2013 dataset.

Research Questions

This section primarily focuses on the method to build a real-time publicly available emotion detection system either be it on a web application or a mobile application. We would try to accomplish our goals for this research by using various freely available python API such as keras, tensorflow etc. We would also use transfer learning technique in order to recognize emotion in real-time which will require implementation of Convolution Neural Network models that are pre-trained and tuned to achieve better results. We would tune these models according to our own needs and specifications.

We would investigate and identify most suitable and efficient methodologies for our emotion recognition system in real-time data stream ?

We would also research into various technical possibilities to create an online available system for emotion classification ?

Research on various data pre-processing techniques in order to gather the best set of data for our research ?

Hypothesis to be tested

One of the most prominent hypothesis this research would be testing is to test the most suitable technologies that ensembles with our goal in order to build a robust, scalable, non-memory constrained and efficiently deployed on easily available cloud environments infrastructures. This would effect in implementing our system to be applied on a Big Data environment.

This research topic interests me as I have been recently researching more about human-machine interaction and various methods to improve the real-time emotions recognition system. Many of the methodologies and techniques has been also taught in the course duration of our Masters degree by specialized professors who focus in specific areas of technologies.

1.1 Computer Vision

It can be defined as the process to transform data from images to a decisive result or a class which helps us classify the results into decisive categories. One of the real life applications of computer vision is to perform image segmentation and transform the data and classify the object within the image obtained from the camera or stored data i.e. images or videos. Identifying facial features and a human face can be mentioned as secondary application of

computer vision. This facial detection feature helps profile identified face and identify the age, gender etc information from the gathered data.

1.2 Face Detection

A human face has some distinguishable features that can differentiate it from other human body parts. Generally a human face can be defined as a object consisting of 2 eyes, 1 nose, lips, eyebrows etc. All these parts and various other muscles perform certain set of movements in order to generate some facial expressions which usually represent an emotion or a sentiment. Psychology can be assumed as a major consumer for emotion detection and face identification. There can be various other applications such as artificial intelligence, digital advertisement, consumer feedback, online gaming, computer-human intelligence, mental health analysis, deep fake detection etc. This research utilizes the Haar Cascade Classifier method. The term Haar denotes a mathematical function (Haar Wavelet) in the form of a box (Zahara et al. 2020). Initial technique of face detection was based on identifying individual pixels RGB value, but these methods would prove to be ineffective in long term and would have to be replaced by efficient and better methods, this is where Haar Cascade Classifier method came into place in order to replace the existing face detection method. This method was proposed by Paul Viola and Michael Jones in respective paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in year 2001. Detection of value of simple features is the base of object detection in this method. Apart from many other reasons to choose features instead of pixels, the most prominent would be that features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data (Viola & Jones 2001). This method processes images into squares, where each box represents several pixels. Individual boxes indicate dark and light areas,this image of the face will automatically be identified according to the face position adjustments to rectify the face of a human.

1.3 Classification of Facial Expressions

Expressions on a human face can be distinguished into two categories i.e. macro-expressions and micro-expressions. While macro-expressions are those which last on a face just for half a second to 4 seconds and are easily noticeable and complement the content of sentiment shown by the person whereas a micro-expression will only last for some fraction of seconds and usually missed by human observers (Tolison 2021). These missed micro-expressions are often missed but mostly reveal a persons accurate feelings about their sentiments and

expressions of what they are trying to convey. This research would try to classify 7 universal different expressions which are described as below:

1.3.1 Anger

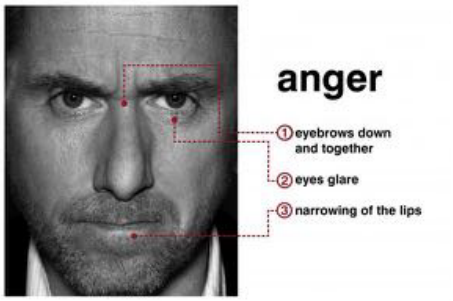


Fig. 1.2 Anger expression (source: (Zahara et al. 2020))

This expression can be displayed by a person when he/she is upset or distressed about another person or a incident and is usually expressed by some facial features such as:

- Pulled together and lowered eyebrows.
- Pressed together lips and corners pointed down or in a square shape.
- Skin between eyebrows form vertical lines.
- Eyes in a continuous hard stare mode.
- Dilated nostrils.

1.3.2 Disgust

Usually a person who is disappointed from others will expression himself with this expression

- Narrowed eyes with eyebrows.
- Raised upper lips.
- Wrinkled nose.
- Raised cheeks.

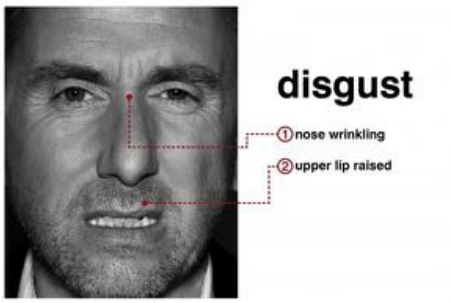


Fig. 1.3 Disgust expression (source: (Zahara et al. 2020))

Fig. 1.4 Fear expression (source: (Zahara et al. 2020))

1.3.3 Fear

- Raised eyebrows along with pushed together.
- Wrinkled eyebrows.
- Only upper white of eyes can be seen.
- Open mouth and tensed lips or stretched lips pulled back.

1.3.4 Happy

- Raised cheeks.
- Lips corner pulled up to make an semi-circle arch.
- A wrinkle from nose to lips.
- Wrinkles besides eyes.
- Low eyelid tensed.

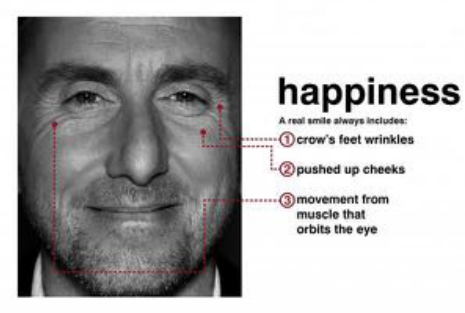


Fig. 1.5 Happy expression (source: (Zahara et al. 2020))

1.3.5 Sadness

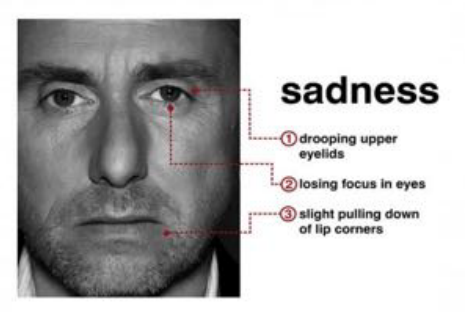


Fig. 1.6 Sad expression (source: (Zahara et al. 2020))

- Eyebrows drawn close together and angled corner upwards.
- Lips corner drawn down.
- Popping out of lower lips.
- Lifting of jaws.

1.3.6 Surprise

- Raised and arched eyebrows.
- Horizontal wrinkles on forehead.
- Open mouth without tension and spacing in teeth.

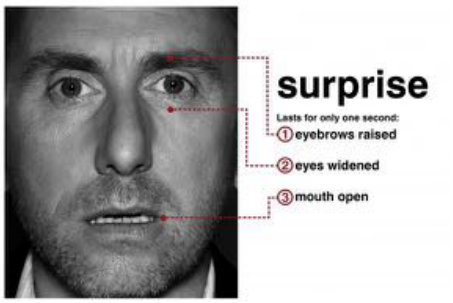


Fig. 1.7 Surprise expression (source: (Zahara et al. 2020))

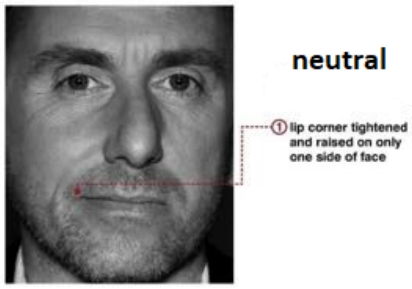


Fig. 1.8 Neutral expression (source: (Zahara et al. 2020))

1.3.7 Neutral

Its a facial expression characterized by neutral positioning of facial features, implying lack of strong emotions (*Blank expression* 2021)). Usually its a sign of lack of any emotions within a person at a certain point of time.

1.4 Deep Learning and Convolution Neural Network

CNN has been proven to be an very efficient method over the past years in pattern recognition and image processing. This algorithm is a part of Deep Neural family because of the high amount of network depth which is significantly multiplied while working with images. It uses Restricted Boltzmann Machine to increase the processing of multiple layers in the network.

1.4.1 Mini XCEPTION

Our system design is based on Xception model, which has been adopted from some of the selected pre-stored architectures in CNN. This model uses convolution pattern and feature extraction by utilizing the additional layers which can be distinguished as different

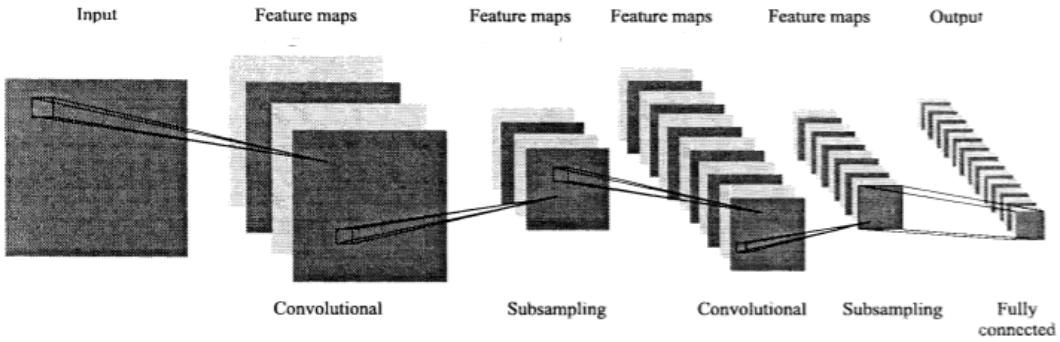


Fig. 1.9 General Convolutional Network Architecture (source: (Albawi et al. 2017))

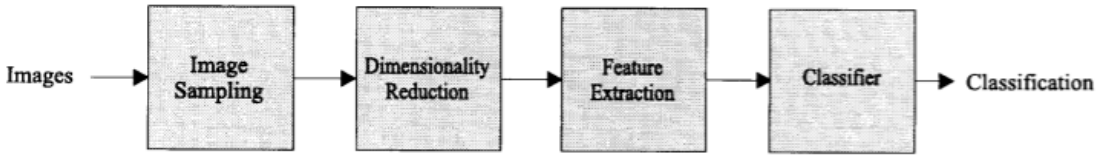


Fig. 1.10 High Level Block Diagram of proposed system (source: (Lawrence et al. 1997))

convolutional layer. While training our data using CNN Mini EXCEPTION model we will be using different techniques such as Data Augmentation, Batch Normalization, Kernel Regularization, Global Average, Pooling and Split Convolution. Using this model we can achieve facial expression classification.

Chapter 2

Related Work and Background

2.1 Aim and Objective

Since many decades research methods have been developed to successfully recognize human emotion. Generally various forms of data can be used to achieve this task which can be audio, text, images, videos etc. This research implied image data in order to identify human emotions from live visual videos and stored images. This involves multiple complicated techniques and processes such as Batch Normalization, Feature Extraction, Feature Selection, Data Augmentation etc. The research aims to test and identify various suitable methodologies and proven techniques in order to classify emotions in real-time video stream. The research can be deployed to a online system which can be used for many for various purposes in real-time. This can be beneficial in machine-human interaction in multiple forms and can be gaming, psychology etc. This research aims to use FER 2013 dataset which is available publicly and training our built deep learning model on this dataset and predict/ classify the emotions onto the new acquired images. Final system targets to classify the emotions of the live streaming data in a successful manner. The proposed system aims the use of Convolutional Neural Network from the Deep Learning family to achieve success.

2.2 Scope and Limitations of the research

The scope of this research has been described by below:

- Design a system for classification of human emotions.
- Utilization of appropriate algorithm from deep learning family to achieve the goal.
- Improve accuracy of the algorithm and results on test dataset.

- Improve the efficiency an d speed of the system.
- Reduce the training time by using available GPU's for the system.
- Explore various functional fields where the system can be implemented.
- Improve the robustness of the Mini-XCEPTION model from Convolutional Neural Network from the Deep Learning family.
- Improve the accuracy of the model from available previous research methods.
- Enhancing the consumer experiences through emotion classification.

The limitations of the research has been listed below:

- Due to memory constraints on the available system, training of the data has been done only on a part of the whole available dataset and also in a restricted multiple layer network.
- The freely available GPU like Google Colab and AWS services used to crash while training the images data.
- Cloud environment were identified as a additional expense to accomplish our system.
- Deployment of such as system for public use can require many permissions from official authorities and governing bodies.

2.3 Literature Review

For past decades many research has been carried out in order to identify the human emotions while using various machine learning and artificial intelligence techniques but each methods proves out to be different and provides altering results. Techniques such as SVM, CNN, DBN, Deep Learning Network has been constantly implemented and improved in order to achieve required results. This techniques provide numerous algorithms and models where each of them would provide distinguish results and altered performances on different datasets.

This research is paralleled and supported by (Lawrence et al. 1997) where researchers have created a automatic system for face recognition with a combination of image samples and SOM network. The KL transformation method proves to be a rapid classification method where each of the images are transformed and classified for better performance and results. Each individual has provided 5 images for various emotions which has significantly brought

down the error margin of face recognition in 3.8% to 10.5% respectively. While rejecting the garbage data from 10% the classification error significantly reduces.

Method for fusion of both audio and video sentiment analysis has been brought out in together in (Han et al. 2014) in order to classify sentiments within and recorded video stream. Researcher Kun Han and his team members worked on various utterance level features and segment level features within audio data stream to classify sentiment in the stream. The accuracy score for this methodology was 54.3% for classifying the emotions where the technique used was ELM classifier. They also achieved over fitting of the results which brought up the accuracy to 90% to 99.5% while using the same classifier technique on the Berlin Emotional Speech Database.

Researcher Eric Chu and Deb Roy from MIT Media Lab have implemented a sentiment analysis system based on audio-visual data while first performing the sentiment recognition on both form of data separately and then using embedding techniques in order to combine the results from both the methods for better accuracy and low error rate in classification (Chu & Roy 2017). This research used AlexNet architecture from deep convolutional neural network to classify sentiment from the images. They used techniques such as batch normalization, PReLU activation unit and ADAM optimization. The model trained on images with sentiment greater than 50% to be marked as "POSITIVE" and less than 50% as "NEGATIVE" (Chu & Roy 2017). The metrics set for this experiment were as follows: learning rate at 0.01 along with batch size of 128 and a batch normalization decay at 0.9. These metrics resulted in a accuracy of 0.652, precision of 0.753, recall at 0.729 and F1 score at 0.741.

In a subsequent research where data was acquired from Youtube open source available data and was considered for sentiment analysis in audio-visual format, researchers Martin, Felix, Tobias and Bjorn from Shanghai Jiaotong University. This dataset consists of 370 videos from Youtube and ExpoTV. Out of these videos 228 videos were marked positive, 23 videos were marked as neutral and 57 as negative. The pre-processing of these videos has been done using a 3D head tracker based on Generalized Adaptive View Appearance Model which computes the features within each video at 30 Hz (Wöllmer et al. 2013). The research uses multi-modal fusion for combining the scores obtained from BLSTM for lingusitic analysis and video features within the video data. The frames within the video alone provides and F1 score of 60.6%. The results and accuracies within this research varied when applied to various databases such as ICT-MMMO, Metacritic etc.

The research (Hossain & Muhammad 2019) uses CNn along with transfer learning while implementing the ImageNet to identify the emotions within acquired static images. In their challenge they used 2015 Emotion recognition sub-challenge dataset of static images which consists various facial expressions. The authors have achieved an accuracy of 55.6%. Initially

they proposed models such as Local Binary Pattern (LBP), Support Vector Machines (SVM) and Guassion Mixture Model (GMM). The ELM based fusion for the eNTERFACE dataset achieved an accuracy of 86.4%.

In a research by R. Cowie and team (Cowie et al. 2001) exploration of facial expression has been carried out with help numerous task where they categorize each type of active and spontaneous expressions in order to classify the emotional state of the person. Their approach was segregated into two sections i.e. gesture oriented and target oriented. While in target oriented approach the expression is decided based on one image of the person at its peak of the emotion whereas in gesture based approach a sequence of multiple images are being captured from a live feed which is then processed and classified for its expressions. In gesture oriented approach each expression usually lasts between 0.5 and 4 seconds. In target oriented approach they use seven 32 X 32 pixel blocks taken from facial regions and use them as a feature to identify the hidden emotions with the facial expression for the person. They acquired an accuracy of around 82% to 87.5% while using various approached along with hardware components for better and quicker processing of data.

In a research by students from Nanyang Technological University in year 2016 (Poria et al. 2016) they implemented convolutional MKL based multimodal emotion recognition system which efficiently approves of the sentiment within a captured video stream. The used USC IEMOCAP database which contains 12 hours of video which has been segregated into 5 minutes of conversation between male and female artists. In this research feature selection was done using CRMKL model from deep CNN module which allowed them to achieve and accuracy of 85.30% within the emotion recognition from the acquired database. The fused other datasets such as textual and audio for this classification purpose. After the fusion of results from all three models and algorithms the accuracy went significantly low. Within their paper they proposed an fusion of audio, text and facial expressions from images while implementing deep convolutional neural network and using multiple kernel learning models.

In a recent published paper they proposed and new feature extraction method call hvnLBP-TOP for video based sentiment extraction (Li & Xu 2019). They also implemented Principal Component Analysis (PCA) and Bidirectional Long Short Term Memory (bi-LSTM) for the purpose of classification and dimensionality reduction. They achieved and accuracy of average recognition of 71.1% on the MOUD database and a accuracy of 63.9% on the CMU-MOSI database. The preprocessing of videos was done by cutting the videos into frames and then stored into jpeg images. Then they used existing face detection API to detect a human face within the picture frame. Then they resized the pictures into 98x98 pixels png format.

Task	MOUD		MOSI			
	Binary		Binary		5-class	Regr.
Metric	Acc.	F1	Acc.	F1	Acc.	MAE
SOTA	67.3	61.2	65.3	54.5	29.5	1.21
Ours	71.1	62.8	63.9	65.9	29.8	1.29

Table 1: Results for MOUD and CMU-MOSI dataset in (source : (Li & Xu 2019)).

The performed sequential learning as well after the features were generated and multiple sequential learning module layers were included such as Bi-LSTM, batch normalization, Two dense layers and dropout layer and a final dense layer for classification purpose. They used SoftMax for classification and sigmoid for regression purpose.

This paper likewise momentarily acquainted a few famous information bases related with FER comprising of both video groupings and still pictures (Ko 2018). In a conventional dataset, human looks have been examined utilizing either static 2D pictures or 2D video groupings. Be that as it may, on the grounds that a 2D-based investigation has trouble taking care of huge varieties in present and unpretentious facial practices, ongoing datasets have thought of 3D looks to more readily work with an assessment of the fine primary changes innate to unconstrained articulations. Moreover, assessment measurements of FER-based methodologies were acquainted with give standard measurements for correlation. Assessment measurements have been broadly assessed in the field of acknowledgment, and accuracy and review are for the most part utilized. Be that as it may, another assessment strategy for perceiving successive looks, or applying miniature articulation acknowledgment for moving pictures, ought to be proposed. Despite the fact that concentrates on FER have been led over the previous decade, as of late the execution of FER has been fundamentally worked on through a mix of profound learning calculations. Since FER is a significant method for imbuing feeling into machines, it is profitable that different investigations on its future application are being directed. On the off chance that passionate situated profound learning calculations can be created and joined with extra Internet-of-Things sensors in the future, it is normal that FER can further develop its present acknowledgment rate, including even unconstrained miniature articulations, to similar level as people.

Chapter 3

Methodology

3.1 Research Methodology

For this research purpose we have used CRISP-DM methodology which stands for Cross-Industry Standard Process for Data Mining. It is an open-source and freely available methodology which is widely used for agile projects by various industry leaders (Wirth & Hipp 2000). It addresses various problems within industry by defining separate process for each of them and combining them into a framework to accomplish an data mining project. It focus on huge data mining projects as well as small scale robust data mining projects as well.

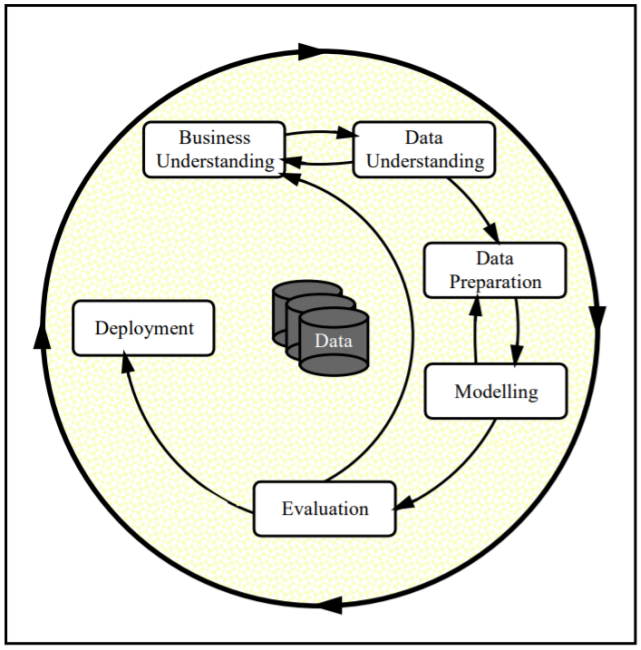


Fig. 3.1 Phases in CRISP-DM (source: (Wirth & Hipp 2000))

It's a generic process model which is used for implementation, planning, documentation and communication within a data mining project. This model can be modified according to specific industry project and is high modifiable. It consists of six phases named as Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment. Within each phase it can be further derived into smaller tasks or phases which will guide through each success full completion of task of the project.

3.2 Research Architecture and Design

Using standard industry methods a system architecture was conceptualized in order to define the structure and behaviour of the proposed system. A framework design fundamentally focuses on the interior points of interaction among the framework's parts or subsystems, and on the interface(s) between the framework and its outer climate, particularly the client. (In the particular instance of PC frameworks, this last, unique, connection point is known as the PC human connection point, AKA human PC point of interaction, or HCI; previously called the man-machine interface.)

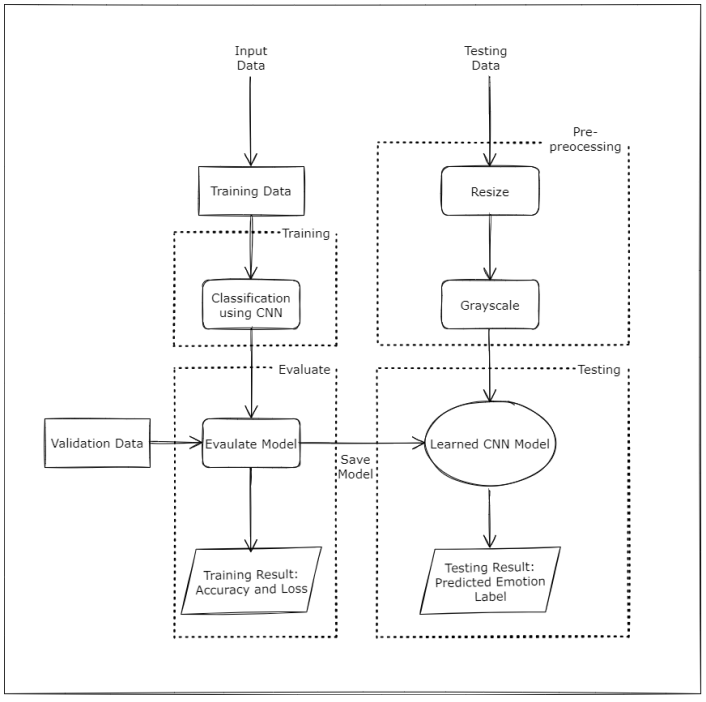


Fig. 3.2 Flowchart of the system architecture

Our initial concept designs would be segregated into 4 main components that are Training, Evaluate, Pre-processing and Testing phases. Within each phase there would be carried out

specific tasks which can be ranged from classification of training data, calculating accuracy and loss, validating data, resizing the data etc. As per our system design we would initially acquire training dataset and after pre-processing it we would pass it for classification using Convolutional Neural Network while applying the mini-Xception architecture model and storing the model results into .hdf5 file which will be later used for on our test data for prediction and classification. The stored architecture model will then be passed onto our testing data which has been already pre-processed i.e. resized and gray-scaled and now ready for classification from the learned CNN model. The learned CNN model would classify or predict the testing or live dataset from one of the 7 categories of the emotions.

3.3 Data Collection

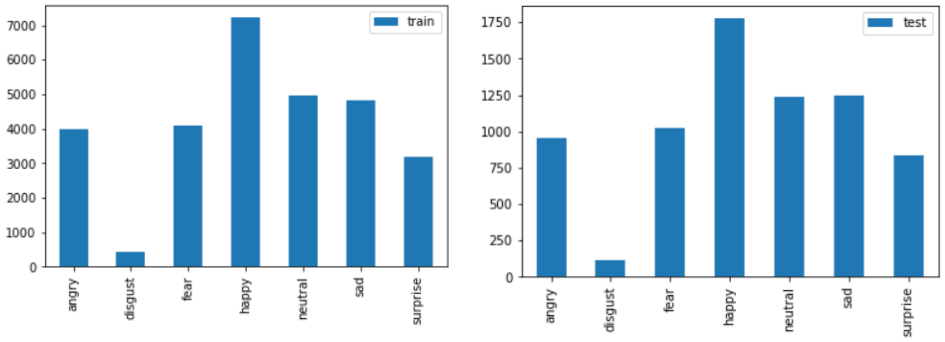
In this research we use secondary data collection method to acquire necessary database. This research is using FER-2013 database which consists of numerous labelled images. This dataset consists 28709 labelled images in the training set and 7187 labelled images in test set (Ozdemir et al. 2019). All the images within this dataset has been labelled to one of the seven emotions: happy, sad, angry, afraid, surprise, disgust and neutral. This dataset consists of both face focused images and non-focused face images which will help our model while training. These images are in a format of grayscale and in a size of 48x48 pixels. This dataset was primarily collected by using google search images for various emotions and different synonyms of its emotions (Ozdemir et al. 2019).

	angry	disgust	fear	happy	neutral	sad	surprise
train	3995	436	4097	7215	4965	4830	3171
	angry	disgust	fear	happy	neutral	sad	surprise
test	958	111	1024	1774	1233	1247	831

Fig. 3.3 FER-2013 images per emotions (source: (Ozdemir et al. 2019))

As per above figure the happy category most number of labelled images whereas the disgust consists of minimal number of images with 600 count. They were captured using an automated system due to which they must have been centered and out of alignment for some of the pictures.

Below bar graph illustrates the distribution of images from various emotions among the FER-2013 dataset. This bar graph shows that there is an uneven distribution of images within some categories namely Disgust. We would have to take care of such factors of uneven dataset bu using techniques such as batch normalization which would prevent the results from being over-fitted or under-fitted.



(a) FER-2013 Training Set Distribution Bar Graph

(b) FER-2013 Test Set Distribution Bar Graph

Fig. 3.4 FER-2013 Training & Test Set Distribution Bar Graph

3.4 Data Pre-processing

Pre-processing of the acquired data has to be taken care in order to avoid issues such as memory crash, wide size and color of the images on various other factors. Design acknowledgment and picture pre-processing can be applied in numerous various regions to take care of existing issues. This is a significant explanation this discipline has become so quick. Thus, different necessities presented during the course of settling useful issues rouse and accelerate the advancement of this discipline.

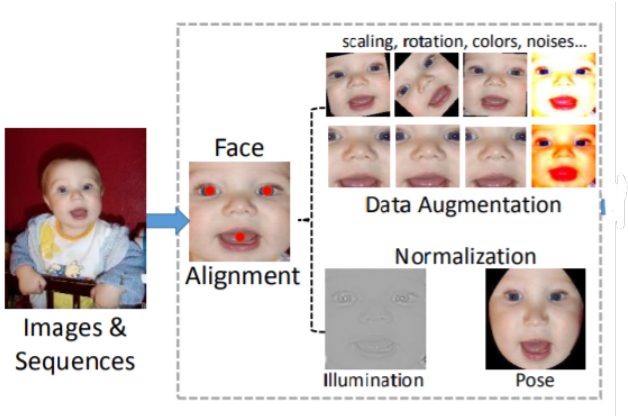


Fig. 3.5 Illustration of pre-processing of data (source: (Li & Deng 2020))

3.4.1 Reshaping Data

The acquired data needs to be converted into 3d tensor which can be further used for training purpose. For this purpose we will be using keras API and tensorflow package which has

been developed for year for such purpose and can are available freely. The tensor consists of various parameters such as . The channel indicates that the images should be in grayscale form. For the pictures in the preparation set, a decent size window (e.g., 5 5) is ventured over the whole picture as displayed and neighborhood picture tests are extricated at each step. At each progression the window is moved by four pixels (Khanzada et al. 2020).

3.4.2 Data Augmentation

It can be defined as an artificial data generator whilst introducing new samples of data created using perturbation method over the original dataset. Its one of the most efficient and frequently implemented method for generating more training data without too much effort in cases where gathering of new data is limited and not feasible. Its been very frequently used in image analysis and augmentation branch of machine learning (Bloice et al. 2017). Since its a widely used technique many developers has developed numerous API which are freely available throughout the industry. After thorough research and experimenting with various commonly used techniques on FER dataset this research has achieved the most promising results while using horizontal mirroring +/- 10 degrees of rotation, +/- 10 degrees of image focus/zoom and +/- 10 degrees of vertical/horizontal shifting. Keras comes with ImageData-Generator class which will allow its users to perform image augmentation in an convenient manner. This class inherits three different methods that are flow(), flow_from_directory() and flow_from_dataframe() which allows us to read images database into huge numpy arrray and various folders consisting images. In this research we will be using flow_from_directory() method for our image augmentation purpose.

3.4.3 Fitting the generator to our data

After image augmentation we will use the batch size as 64 for fitting out data into our image generator. Hence data will be computed in batch size of 64. The best way to train such huge amount of data will be to use data generators while working with images (*Facial Emotion Detection Using CNN* 2021). Within the Keras API, the ImageDataGenerator class defines configurations of image data preparation and augmentation. The various functionalities within this class are Sample-wise standardization, Feature-wise standardization, ZCA whitening, Random rotation/shifts/shear/ flips, Dimensionality reduction, Saving of augmented images to disk etc. Instead of applying various functionalities on whole image dataset within the storage, this API is developed to be iterated by deep learning model fitting process (Brownlee 2019). Once the configuration of ImageDataGenerator has been done one must fit the image

data. This helps us perform all the statistics required to implement the tranforms of the images. This can be carried out using the fit() function on data generator an dtraining data.

3.4.4 Class Weighting

In order to leverage class imbalance problem within the various categories in our database we inherited class weighting inversely proportional to the total amount of samples within the dataset. Within our Disgust category we successfully reduced the misclassification rate from 61%.

3.4.5 SMOTE - Synthetic Minority Over-sampling Technique

This technique was used in order to perform oversampling over the minority classes within the database and under-sampled classes to achieve the best performance and result. Since this technique sometimes perfectly balances all the classes within the database this can produce another problem of overfitting of the training dataset and hence forth we decided to experiment further and use various other techniques.

3.4.6 Face Alignment

In this research we enlist some out performing face alignment techniques that have been widely used in deep FER. Provided the FER dataset, we have to detect the face from training data and later deduct the background and areas without any human face. We use Haar Cascade Face Detection API in our research to detect and focus on the face within the image which has been proven robust and simple in computation for identification of human faces in and image frame or video. Among with face detection as one of the indispensible procedure which enables our feature learning we will also need to carry face alignment to enhance the performance of FER dataset (Li & Deng 2020). Both of these pre-processing steps will significantly reduce the issues such as variation in face size and in-place rotation. In recent times cascaded regression in combination with deep neural network has been a prominent player in face alignment because of its high accuracy and speed. Some research has also implemented multiple combined face alignment techniques in order to acquire results for processing human faces.

3.4.7 Face Normalization

Varieties in brightening and head stances can present huge changes in pictures and consequently disable the FER execution. Accordingly, we present two average face standardization strategies to improve these varieties:

- Illumination normalization
- Pose normalization

Illumination normalization

Illumination and difference can shift in various pictures even from a similar individual with the equivalent articulation, particularly in unconstrained conditions, which can bring about enormous intraclass differences. Different calculations, such as isotropic diffusion (IS)- based normalization, discrete cosine transform (DCT)- based normalization, difference of Gaussian (DoG) and homomorphic filtering based standardization, can be utilized for illumination normalization. In addition, related studies have shown that histogram equalization joined with illumination normalization brings about better face acknowledgment execution than that accomplished utilizing illumination normalization alone. Many investigations in the writing of profound FER have utilized histogram equalization to increment the worldwide differentiation of pictures for preprocessing. This strategy is successful when the brightness of the background and foreground are comparable. In any case, straightforwardly applying histogram equalization may overemphasize nearby contrast. To take care of this issue proposed a weighted summation way to deal with consolidate histogram equalization and linear mapping(Li & Deng 2020).

Pose normalization

Pose variety is another normal and recalcitrant issue in unconstrained settings. A few examinations have utilized Pose normalization methods to yield front facing facial views for FER among which the most well known was proposed by Hassner et al. In particular, subsequent to localization facial spots, a 3D surface model conventional to all faces is created to appraise apparent facial parts. Then, at that point, the underlying frontalized face is orchestrated by backprojecting each information face picture to the reference coordinate framework. Then again, Sagonas et al. proposed a measurable model that all the while confines tourist spots and converts facial stances utilizing just front facing faces. As of late, a progression of GAN-based profound models were proposed for front facing view amalgamation and announced promising exhibitions (Li & Deng 2020).

3.5 Feature Extraction

The newly introduced models such as ResNet and DenseNet had a significant improvement over one of the most prominent problem in deep network which was prone to gradient disappearance in backpropogation (Chang et al. 2018). This newly introduced models also brought enhancements into classification of images. This research proposes CNN for feature extraction of within the image dataset. We propose and implement the mini Xception model to classify the images and emotions within this dataset. Methods such as use of residual modules and depth-wise separable convolutions. Residual modules works to enhance the desired mapping between two consecutive layers in order that learned features become outcomes as the difference in between feature map and desired features (Chang et al. 2018). Desired features $H(x)$ is modified in order to simplify training problem $F(x)$ as shown below:

$$H(x) = F(x) + x$$

The parameter reduction takes place within this research from convolutional layers which has been carried out by using depth-wise separable convolutions. Depth-wise separable convolutions is made up of different layers i.e. depth-wise convolutions and point wise convolutions.

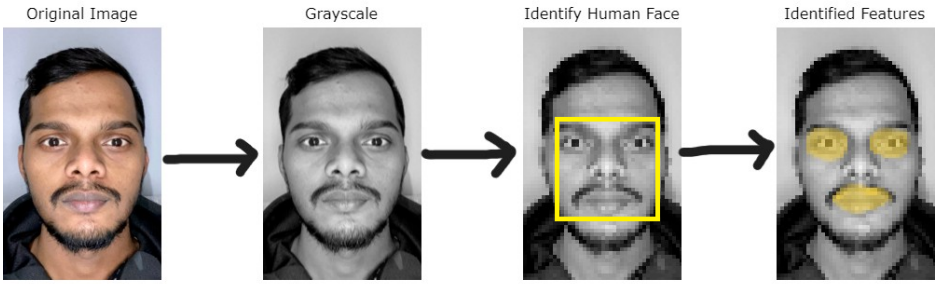


Fig. 3.6 Feature extraction process for proposed model

This research used the feature extraction using CNN by which the input of the FER database for each feature value $f_i(i = 1,...,n)$ where f_i will be the i-th feature extracted from acquired image (Chang et al. 2018). Feature extraction from the training dataset was randomly segregated $X = [x_1, x_2, ..., x_n]$ which were further anonymously divided into K folds, where $x_i \in R_d, i = 1,2,...,n$ d can be mentioned as the feature dimension of each sample image and x_i can be said as the i-th sample feature vectors. First we chose the fold of image samples for training set and then the $K-1$ folds have been picked for test set in order to enhance the generalization of base classifier and distinguish classifiable images. Our approach results in K base classifiers on various training sets. This process was repeated

for numerous m times to verify $N(N = KM)$ trained base classifier. After which all the training sample feature vectors are predicted by n base classifiers, following with counting the appropriate number of prediction calculation categories and computing the degree of classification $R(x_i)$ for i-th sample.

$$R(x_i) = N(x_i)/N$$

here,
n - number of base classifiers
$N(x_i)$ - appropriate classification number of N base classifier of i-th training sample.

Our methodology evaluates the complexity of sample classification features by ease degree of classification $R(x_i)$. We also conjugate parameters named ease threshold θ which is defined as the boundary to separate the easily classifiable sample from difficult ones.

$$x_i \in \begin{cases} S_E & \text{if } R(\mathrm{x}_i) \geq \theta \\ S_D & \text{if } R(\mathrm{x}_i) < \theta \end{cases}$$

Fig. 3.7 Formula for ease degree of classification for i-th training sample

In above formula if the ease of degree of classification for i-th training sample $R(x_i) \geq \theta$, we will be dividing it into easy classification sample subsets (S_E, L_E). We also divided it into complex classification sample subspace (S_D, L_D) whilst the ease of degree of classification for i-th training sample $R(x_i) < \theta$ (Fatima et al. 2021).

Some of the challenges that were taken care while feature extraction process can described below:

- The background noise within the sample images were reduced due to use of CNN models and heavy computation.
- Resizing the sample image database to appropriate size as input to the training model.
- Grayscaling of the images within the database where some colours were visible.
- Correcting the appropriate positioning of the faces within the database in order to capture the facial features.
- obliteration of the sample images where the quality was too poor.

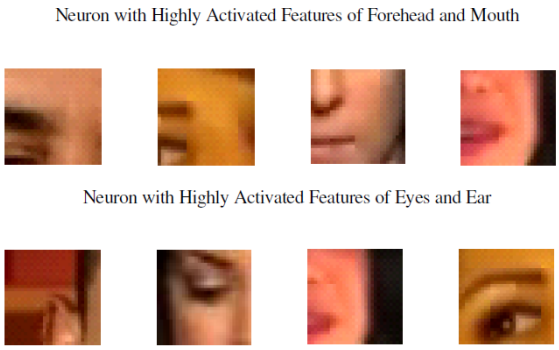


Fig. 3.8 Most selected image segments at feature detector within layers of CNN

3.6 Model Architecture

Our proposed model were assessed in understanding to their test precision and number of parameters. The model were planned with making the best accuracy over number of parameters proportion. Decreasing the quantity of parameters help us beating two significant issues. To begin with, the utilization of little CNNs ease us from slow performance in equipment compelled frameworks. Also second, the decrease of parameters gives a superior generalization under an Occam's razor framework. Our model depends on dispensing with fully connected layers.

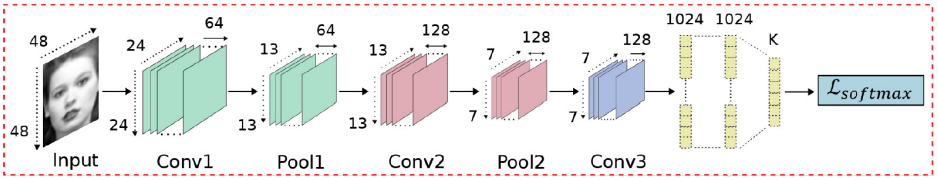


Fig. 3.9 Convolution model architecture

The subsequent design consolidates the erasure of the completely connected layer and the consideration of the combined depth-wise distinguishable convolutions and residual modules. Models were prepared with the ADAM optimizer. Following the past design architectures, our underlying architecture utilized Global Average Pooling to totally eliminate any fully connected layers. This was accomplished by having in the last convolutional layer a similar number of feature maps as number of classes, and applying a softmax activation function. Our proposed model for real-time classification. to each decreased feature map. Our underlying proposed engineering is a standard completely convolutional neural network made out of 3 convolution layers consisting of ReLUs, Batch Normalization also Global Average Pooling. This model contains roughly 2400000 parameters. It was prepared on the FER 2013 database, which contains 300000 pictures where each picture has a place with the

classified emotions, and it accomplished a precision of 98.5% in this training dataset. Our underlying model accomplished an accuracy of 62.5% in this test/validation dataset. We will allude to this model as "sequential fully-CNN".

Since our underlying proposed design erased the last completely associated layer, we diminished further how much parameters by dispensing with them now from the convolutional layers. This was done by the utilization of depth-wise separable convolutions. Depth-wise separable convolutions are formed of two distinct layers: depth-wise convolutions and point-wise convolutions. The fundamental motivation behind these layers is to separate the spatial cross-correlations from the channel cross-correlations. They do this by first applying a DD filter on each M information channels and afterward applying $N11M$ convolution channels to join the M information channels into N output channels. Applying $11M$ convolutions joins each worth in the feature map disregarding their spatial connection inside the channel. Depth-wise convolutions diminishes the calculation as for the standard convolutions by a component of $1/N + 1/D_2$. A representation of the contrast between an ordinary Convolution layer and a depth-wise convolution can be seen. Our design is a completely convolutional neural network that contains 3 residual depth-wise separable convolutions where every convolution is trailed by a batch normalization activity and a ReLU activation function. The last layer applies a global average pooling and a softmax activation capacity to create an prediction. This designs gets an exactness of 92.5% in emotion recognition task within the training set. Which compares to a decrease of one percent regarding our initial execution. By decreasing our designs computational expense we are currently ready to join use them successively in a similar picture without any genuine time decrease. Our total pipeline including the openCV face location module, the emotion grouping also the feeling order takes 0.220.0003 ms on a i5-4210M CPU. This compares to a speedup of 1.5× when contrasted with the first design. We additionally added to our execution an ongoing directed back-propagation visualization to see which pixels in the picture enact a component of a more significant level feature map. Given a CNN with just ReLUs as activation functions for the transitional layers, directed back propagation takes the derivative of each component (x,y) of the input picture I with regard to a component (i, j) of the feature map f^L in layer L (Arriaga et al. 2017). The upcoming newly generated image R segregates all negative gradient, we select the leftover gradient so that there is only enhancement in the value of selected element of feature map. Within the research a fully connected ReLU CNN is again constructed picture in layer l can be described as below:

The proposed real-time emotion recognition system pipeline consists of components such as face detection, emotion classification and have been fully integrated into the final system. Sometimes while training the neural networks on minute facial expression brings up issues

$$R_{i,j}^{l} = (R_{i,j}^{l+1} > 0) * R_{i,j}^{l+1}$$

Fig. 3.10 ReLU CNN reconstructed image

such as overfitting. Inspite of directly using the available pre-trained and fine-tuned model to extract feature on FER dataset, we propose to design a multi-stage fine tuning strategy in order to acquire best performance and results.

The conventional softmax loss layer in CNNs essentially powers features of various classes to stay separated, however FER in sensible conditions experiences high interclass similarity as well as high intraclass variety. Subsequently, a few works have proposed novel misfortune layers to mitigate this issue. Enlivened by the center loss, which penalizes the distance between deep feature and their comparing class focuses, two varieties were proposed to help the management of the softmax loss for more discriminative features. Island loss was formalized to build the pairwise distances between various class centers. In particular, the island loss determined at the feature extraction layer and the softmax loss determined at the decision layer are consolidated to regulate the CNN training. Locality-preserving loss (LP loss) was formalized to pull the locally adjoining features of a similar class together so that the intraclass neighborhood clusters can be nearer for every articulation. Mutually training this loss with the softmax loss, the discriminative influence of the learned feature can be exceptionally improved. In light of the triplet loss, which requires one positive example to be nearer to the anchor than one negative example with a fixed gap, two varieties were proposed to replace or help the management of the softmax loss. Remarkable trio based loss was formalized to give troublesome examples more weight when refreshing the network. $(N+M)$- tuples cluster loss was formalized to mitigate the trouble of anchor selection and threshold validation in the triplet loss for identity invariant FER. During training, identity-aware hard-negative mining and online positive mining schemas were utilized to diminish the inter-identity variation in a similar articulation (Li & Deng 2020).

3.7 Hyperparamter Tuning

Since our proposed model is ready to learn patterns from the FER dataset within the training data which can be further used for emotion classification. We can perform hyperparameter tuning in order to enhance the performance of our network which will reduce issues such as overfitting and underfitting. Our proposed model uses an image of size 48x48 for classification purpose. We set the number of channel to default settings and use the Adam

Table 3.1 Hyperparameter values used in Xception model

Xception Model parameter	Value
Input image size	48 X 48
Number of channels	Default
Optimizer	Adam
Learning Rate	0.0001
Batch Size	128
Epochs	50

optimizer. We set the learning rate to 0.0001 alon with a batch size of 128 for training and testing purpose. Our system uses 50 epoch for improvement in model accuracy and loss validation along with an early stopping attached method.

Table 3.1 will summarize the hyperparameter values used in Xception model for our proposed approach.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 46, 46, 32)	320
conv2d_1 (Conv2D)	(None, 44, 44, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 22, 22, 64)	0
dropout (Dropout)	(None, 22, 22, 64)	0
conv2d_2 (Conv2D)	(None, 20, 20, 128)	73856
max_pooling2d_1 (MaxPooling2D)	(None, 10, 10, 128)	0
conv2d_3 (Conv2D)	(None, 8, 8, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 128)	0
dropout_1 (Dropout)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1024)	2098176
dropout_2 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 7)	7175

Total params: 2,345,607
Trainable params: 2,345,607
Non-trainable params: 0

Fig. 3.11 Parameter selection by CNN model

3.8 Classification Accuracy and Loss Validation

Below figure displays the classification accuracy and validation loss for our training and test datasets. As the number of epoch increase we see significant decrements in our validation loss and a visible increment in classification accuracy. Once the training loss and validation loss along with each number of epoch decreases our model hustles to recognize pattern in the image data. During the training process the model tries to find the optimum value for the accuracy value i.e. train and validation accuracy along with the number of epochs.

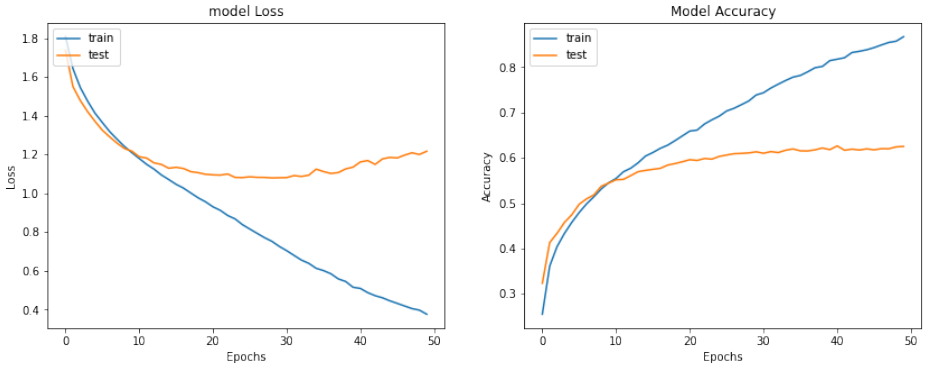


Fig. 3.12 Classification Accuracy and Validation Loss

```
449/449 [==============================] - 45s 99ms/step - loss: 0.1310 - accuracy: 0.9825
113/113 [==============================] - 11s 97ms/step - loss: 1.2170 - accuracy: 0.6250
final train accuracy = 98.25 , validation accuracy = 62.50
```

Fig. 3.13 Training Accuracy and Validation Accuracy

Once irrelevant data was discarded in previous phases of the research and we only used the relevant data and images we found out that the training dataset and test datasets losses and accuracies are reduced which can be seen in above figure. Accuracy within the training dataset and the training loss has significantly decreased while the validation accuracy and loss has significantly improved (Kim et al. 2020).

Chapter 4

Results

This section we are describing our findings and outputs for the classification problem we undertook with our research.

4.1 Evaluation Metrics

Methods of evaluation metrics within the FER dataset can be classified into 4 categories from various attributes:

Precision

It is denoted by P. It can be defined as the $TP/(TP+FP)$. It is the fraction of automation annotations of emotion i which have been appropriately identified (Ko 2018).

Recall

It is denoted by R. It can be defined as the $TP/(TP+FN)$. It is the amount of appropriate recognition of emotions i over the actual number of images with emotion i (Ko 2018).
Here,
TP - number of True Positive within the datasets.
FN - number of False Negative within the datasets.
FP - number of False Positive within the datasets.

Accuracy

It is denoted as ACC. It can be defined as the ratio of true outcomes (True Positive to True Negative) with the total number of cases examined (Ko 2018).

$$Accuracy(ACC) = (TP + TN)/TotalSample$$

F1-score

It is divided into two metrics which depends on if they use spatial or temporal data: frame-based F1-score and event-based F1-score (Ko 2018). Individual metrics stores various properties of the output. It has a predictive power in terms of spatial consistency, whereas event-based F-score has predictive power in terms of temporal consistency (Ko 2018).

$$F1 - frame = 2RP/(R + P)$$

$$F1 - event = (2ERXEP)/(ER + EP)$$

Here,
ER - event-based recall EP - event-based precision
Below tables illustrates the classification report for both training and test dataset.

Classification Report

	precision	recall	f1-score	support
angry	0.14	0.15	0.14	3995
disgust	0.02	0.02	0.02	436
fear	0.14	0.13	0.14	4097
happy	0.25	0.25	0.25	7215
neutral	0.18	0.18	0.18	4965
sad	0.17	0.17	0.17	4830
surprise	0.11	0.11	0.11	3171

(a) Classification Report for Training Dataset

Classification Report

	precision	recall	f1-score	support
angry	0.12	0.12	0.12	958
disgust	0.02	0.02	0.02	111
fear	0.13	0.11	0.12	1024
happy	0.25	0.27	0.26	1774
neutral	0.16	0.16	0.16	1233
sad	0.18	0.20	0.18	1247
surprise	0.11	0.10	0.11	831

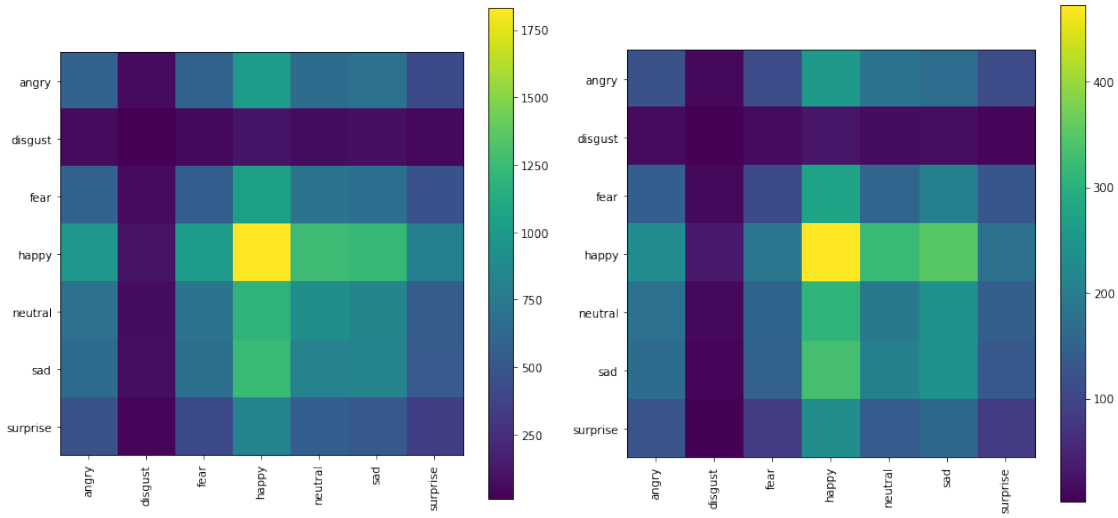
(b) Classification Report for Test Dataset

Fig. 4.1 Classification Report for Training & Test Dataset

4.2 Confusion Matrix

It can be derived as the table for the summarizing various aspects of performance of our classification algorithm. The confusion matrix helps us to understand the various types of error within our classification problem. IT briefly summarizes the prediction results for our classification problem for the test as well as training dataset.

From the above confusion matrix, we can deduce that our model has a high success rate while classifying happy people and it has low success rate while classifying disgust emotions



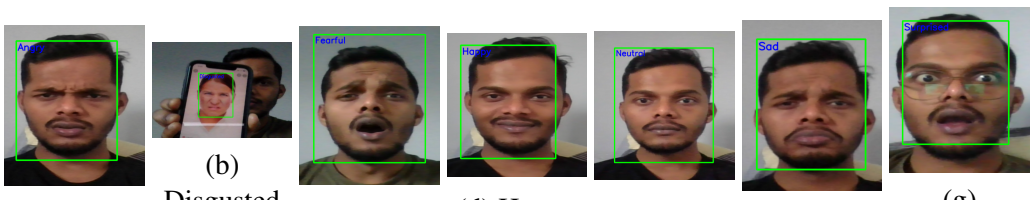
(a) Confusion Matrix for Training Dataset (b) Confusion Matrix for Test Dataset

Fig. 4.2 Confusion Matrix for Training & Test Dataset

within both the training and test dataset. The low success rate for classifying the disgust emotion would be due to less training data for disgust class.

4.3 Live Prediction

Our systems implies the use of CNN architecture model in order to facial expression detection which can be implemented optimally in real-time detection. Our research aims to fabricate a system to recognize the face and identify the facial emotions. The success rate in recognizing the emotion in test dataset is 62.5% which concludes that our proposed system gives good results. For our real-time system within a live video stream each the system recognizes the emotions for each frame. There have been some difficulty to identify the Disgust emotions within the live stream.



(a) Angry (b) Disgusted (c) Fear (d) Happy (e) Neutral (f) Sad (g) Surprised

Fig. 4.3 Testing emotions in real-time machine

Chapter 5

Conclusion

Our research proposes and tests of building a high level design of real-time emotion recognition system based on Convolutional Neural Network. This system was built in order to bring down the speed and processing power any emotion recognition system. While building this system we have successfully eliminated and deduced the amount of parameters used within our model via using depth-wise separable convolutions. We have displayed our capability of detecting and classifying multiple faces and multiple emotions at the same time in a real-time video stream system.This research has achieved the human level performance for the classification of the emotions using the mini-Xception within the CNN which will ultimately leverage the recent architecture constructed by other researchers. We have also absorbed that we need huge of amount of images within the FER dataset benchmark in order to improve performance in emotion recognition.

We observed that huge datasets can be standardized using methods such as data standardization, data cleaning, data pre-processing techniques which has been proposed in this paper which will significantly reduce efforts and time for cropping and resizing of huge amount of image datasets. We also inherited that image augmentation techniques provides us effortless artificial data generation technique to withstand problems such as over sampling or under sampling which can be carried out by using various freely available API's within the industry. Our research investigates into many other available research publications in order to cover and explore all major deep learning techniques used for facial emotion recognition which has been compared from different aspects. Our research demonstrates that the FER emotion recognition model can be registered in real world with developing into any sort of web application. Our research propose the use of mini-Xception an enhanced model over Xception model which will use the residual networks for emotion classification and recognition.

Our convolution newtwork has achieved a total number of 2,345,607 parameter after feature selection process. It has also acquired and 98.25% accuracy for Training dataset and 62.50% for the Testing dataset. As the number of epochs increase the value of model accuracy increases whereas the model loss decreases. We have performed accurate classifications instead of Disgust emotion which has performed very low accuracy while classifying the same emotion due to less amount of training images within our dataset.

5.1 Future Scope

The future work will focus on expanding functionality to mirror the augmentation of reference image data or replicating the advanced methods of pre-processing and augmentation methods such as data manipulation technique. One of the future scope for this research would be to move from unimodal data classification to multi-modal data classification which can take care of complex outputs. Within our research we proposed an novel deep learning technique in order to classify emotions within images. Our future focus would be improve the accuracy of the emotion detection algorithm and decrease the validation loss in our model. We propose and plan to research this dataset further research as it can hold onto challenging and technical tasks which can have shorter and quicker output generation into our image dataset. The rush into recent research over the convolutional neural network implies that it is capable of executing the issue of emotion recognition while conserving higher low-level and short-term discriminative capabilities. Our future scope and research implies us to design an cloud based architecture for emotion classification purpose. We would also propose to optimize the normalization process of the images, rotation and scaling of the image. The current training process require huge time for the training of the model, future research will try to reduce the time, memory consumption and efforts put into this task.

References

Albawi, S., Mohammed, T. A. & Al-Zawi, S. (2017), Understanding of a convolutional neural network, *in* '2017 International Conference on Engineering and Technology (ICET)', Ieee, pp. 1–6.

Arriaga, O., Valdenegro-Toro, M. & Plöger, P. (2017), 'Real-time convolutional neural networks for emotion and gender classification', *arXiv preprint arXiv:1710.07557* .

Blank expression (2021).
URL: *https://en.wikipedia.org/wiki/Blank$_e$xpression*

Bloice, M. D., Stocker, C. & Holzinger, A. (2017), 'Augmentor: an image augmentation library for machine learning', *arXiv preprint arXiv:1708.04680* .

Brownlee, J. (2019), 'Image augmentation for deep learning with keras'.
URL: *https://machinelearningmastery.com/image-augmentation-deep-learning-keras/*

Chang, T., Wen, G., Hu, Y. & Ma, J. (2018), 'Facial expression recognition based on complexity perception classification algorithm', *arXiv preprint arXiv:1803.00185* .

Chu, E. & Roy, D. (2017), Audio-visual sentiment analysis for learning emotional arcs in movies, *in* '2017 IEEE International Conference on Data Mining (ICDM)', IEEE, pp. 829–834.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W. & Taylor, J. (2001), 'Emotion recognition in human-computer interaction', *IEEE Signal Processing Magazine* **18**(1), 32–80.

Facial Emotion Detection Using CNN (2021).
URL: *https://www.analyticsvidhya.com/blog/2021/11/facial-emotion-detection-using-cnn*

Fatima, S. A., Kumar, A. & Raoof, S. S. (2021), Real time emotion detection of humans using mini-xception algorithm, *in* 'IOP Conference Series: Materials Science and Engineering', Vol. 1042, IOP Publishing, p. 012027.

Han, K., Yu, D. & Tashev, I. (2014), Speech emotion recognition using deep neural network and extreme learning machine, *in* 'Interspeech 2014'.

Hossain, M. S. & Muhammad, G. (2019), 'Emotion recognition using deep learning approach from audio–visual emotional big data', *Information Fusion* **49**, 69–78.

Khanzada, A., Bai, C. & Celepcikay, F. T. (2020), 'Facial expression recognition with deep learning', *arXiv preprint arXiv:2004.11823* .

Kim, J. H., Mutegeki, R., Poulose, A. & Han, D. S. (2020), 'A study of a data standardization and cleaning technique for a facial emotion recognition system', pp. 1193–1195.

Ko, B. C. (2018), 'A brief review of facial emotion recognition based on visual information', *sensors* **18**(2), 401.

Lawrence, S., Giles, C., Tsoi, A. C. & Back, A. (1997), Face recognition: a convolutional neural-network approach, *in* 'IEEE Transactions on Neural Networks', Ieee, pp. 98–113.

Li, H. & Xu, H. (2019), Video-based sentiment analysis with hvnlbp-top feature and bi-lstm, *in* 'proceedings of the AAAI conference on artificial intelligence', Vol. 33, pp. 9963–9964.

Li, S. & Deng, W. (2020), 'Deep facial expression recognition: A survey', *IEEE transactions on affective computing* .

Ozdemir, M. A., Elagoz, B., Alaybeyoglu, A., Sadighzadeh, R. & Akan, A. (2019), Real time emotion recognition from facial expressions using cnn architecture, *in* '2019 medical technologies congress (tiptekno)', IEEE, pp. 1–4.

Poria, S., Chaturvedi, I., Cambria, E. & Hussain, A. (2016), Convolutional mkl based multimodal emotion recognition and sentiment analysis, *in* '2016 IEEE 16th International Conference on Data Mining (ICDM)', pp. 439–448.

Tolison, M. (2021), 'Social skills: Understanding the micro and macro expressions'.
URL: *https://dandelionfamilycounseling.com/2020/08/02/social-skills-understanding-the-micro-and-macro-expressions/*

Viola, P. & Jones, M. (2001), Rapid object detection using a boosted cascade of simple features, *in* 'Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001', Vol. 1, Ieee, pp. I–I.

Wirth, R. & Hipp, J. (2000), Crisp-dm: Towards a standard process model for data mining, *in* 'Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining', Vol. 1, Springer-Verlag London, UK, pp. 29–39.

Wöllmer, M., Weninger, F., Knaup, T., Schuller, B., Sun, C., Sagae, K. & Morency, L.-P. (2013), 'Youtube movie reviews: Sentiment analysis in an audio-visual context', *IEEE Intelligent Systems* **28**(3), 46–53.

Zahara, L., Musa, P., Wibowo, E. P., Karim, I. & Musa, S. B. (2020), The facial emotion recognition (fer-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (cnn) algorithm based raspberry pi, *in* '2020 Fifth International Conference on Informatics and Computing (ICIC)', IEEE, pp. 1–9.

Appendix A

Appendix B consists of technical specifications required to implement this research.

Technical Specifications and Requirement

System Requirements

- 8 GB RAM (Minimum)
- 1 TB HDD / SDD
- MUlti-core processor
- Microsoft Windows 7 or higher

Tools

- Microsoft Visual Studio Code
- Notepad++ Editor
- Anaconda Navigator
- Lucidchart
- Jupyter Notebook
- PyCharm IDE

Programming Languages

- python programming
- Latex

Appendix B

Appendix B consists of python code implementation for purpose such as training, evaluation of our model.

```
# import required packages
#import cv2 package
import cv2
#import numpy package
import numpy as np
#import pandas package
import pandas as pd
#import os package
import os
#import seaborn package
import seaborn as sns
#import tensorflow package API
import tensorflow as tf
#import keras package and its necessary modules for training purpose
import keras
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dense, Dropout, Flatten
from tensorflow.keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
from keras.models import model_from_json
from tensorflow.keras.utils import plot_model
# import matplotlib package for plotting purpose
import matplotlib.pyplot as plt
```

```
# Initialize image data generator with rescaling
# Normalization
train_data_gen = ImageDataGenerator(rescale=1./255)
validation_data_gen = ImageDataGenerator(rescale=1./255)
```

```
# initializing training and testing path
train_dir = "data/train/"
test_dir = "data/test/"

# define function to read and count the number of images into the path for each
# class/emotion
def count_exp(path, set_):
    dict_ = {}
    for expression in os.listdir(path):
        dir_ = path + expression
        dict_[expression] = len(os.listdir(dir_))
    df = pd.DataFrame(dict_, index=[set_])
    return df

# execute the function for counting the images within training set
train_count = count_exp(train_dir, 'train')
# execute the function for counting the images within test set
test_count = count_exp(test_dir, 'test')
# print number of images within each class of training set
print(train_count)
# print number of images within each class of test set
print(test_count)
```

```
# plot a bar graph for all classes within training set
train_count.transpose().plot(kind='bar')
# plot a bar graph for all classes within test set
test_count.transpose().plot(kind='bar')
```

```
# Process training data into bacthes of augmented data using the
# flow_from_directory function into keras API
train_generator = train_data_gen.flow_from_directory(
        'data/train',
        target_size=(48, 48),
        batch_size=64,
        color_mode="grayscale",
        class_mode='categorical')
```

```
# Process test data into bacthes of augmented data using the
# flow_from_directory function into keras API
validation_generator = validation_data_gen.flow_from_directory(
```

```
        'data/test',
        target_size=(48, 48),
        batch_size=64,
        color_mode="grayscale",
        class_mode='categorical')
```

```
# create model structure
emotion_model = Sequential()
# Block 1
# convolutional layer 1
emotion_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
                    input_shape=(48, 48, 1)))
emotion_model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

# Block 2
# convolutional layer 2
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

# Block 3
# convolutional layer 3
emotion_model.add(Flatten())
emotion_model.add(Dense(1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(7, activation='softmax'))

cv2.ocl.setUseOpenCL(False)

emotion_model.compile(loss='categorical_crossentropy',
                    optimizer=Adam(lr=0.0001, decay=1e-6), metrics=['accuracy'])
```

```
# Summarize the computed model
emotion_model.summary()
```

```
# Train the neural network/model
emotion_model_info = emotion_model.fit(
```

```
        train_generator,
        steps_per_epoch=28709 // 64,
        epochs=50,
        validation_data=validation_generator,
        validation_steps=7178 // 64)
```

```
# Plot the accuracy, validation accuracy, loss and validation loss for training and
# test dataset
print(emotion_model_info.history.keys())
plt.figure(figsize=(14,5))
plt.subplot(1,2,2)
plt.plot(emotion_model_info.history['accuracy'])
plt.plot(emotion_model_info.history['val_accuracy'])
plt.title('Model Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend(['train', 'test'], loc='upper left')

plt.subplot(1,2,1)
plt.plot(emotion_model_info.history['loss'])
plt.plot(emotion_model_info.history['val_loss'])
plt.title('model Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
plt.savefig("modelloss _modelaccuracy.png")
```

```
# Print accuracy for training set
train_loss, train_accu = emotion_model.evaluate(train_generator)
# Print accuracy for test set
test_loss, test_accu = emotion_model.evaluate(validation_generator)
print("final train accuracy = {:.2f} ,
        validation accuracy = {:.2f}".format(train_accu*100, test_accu*100))
```

```
# write model structure in json file
model_json = emotion_model.to_json()
# use the write method to write our json weights
# using with loop to write the file
with open("emotion_model.json", "w") as json_file:
    # write the json file
    json_file.write(model_json)
```

```
# save trained model weight and the modek in .h5 file
emotion_model.save_weights('emotion_model.h5')
emotion_model.save("model.h5")
```

```
# Confusion Matrix and Classification on training set
y_pred = emotion_model.predict(train_generator)
y_pred = np.argmax(y_pred, axis=1)
class_labels = validation_generator.class_indices
class_labels = {v:k for k,v in class_labels.items()}

from sklearn.metrics import classification_report, confusion_matrix
cm_train = confusion_matrix(train_generator.classes, y_pred)
print('Confusion Matrix')
print(cm_train)
print('Classification Report')
target_names = list(class_labels.values())
print(classification_report(train_generator.classes, y_pred,
        target_names=target_names))

plt.figure(figsize=(8,8))
plt.imshow(cm_train, interpolation='nearest')
plt.colorbar()
tick_mark = np.arange(len(target_names))
_ = plt.xticks(tick_mark, target_names, rotation=90)
_ = plt.yticks(tick_mark, target_names)
plt.savefig("confusion_matrix_train.png")
```

```
# Confusion Matrix and Classification on test set
y_pred = emotion_model.predict(validation_generator)
y_pred = np.argmax(y_pred, axis=1)
class_labels = validation_generator.class_indices
class_labels = {v:k for k,v in class_labels.items()}

#from sklearn.metrics import classification_report, confusion_matrix
cm_test = confusion_matrix(validation_generator.classes, y_pred)
print('Confusion Matrix')
print(cm_test)
print('Classification Report')
target_names = list(class_labels.values())
print(classification_report(validation_generator.classes, y_pred,
        target_names=target_names))
```

```
15
16 plt.figure(figsize=(8,8))
17 plt.imshow(cm_test, interpolation='nearest')
18 plt.colorbar()
19 tick_mark = np.arange(len(target_names))
20 _ = plt.xticks(tick_mark, target_names, rotation=90)
21 _ = plt.yticks(tick_mark, target_names)
22 plt.savefig("confusion_matrix_test.png")
```

Appendix C

Appendix C consists of python code testing our model in real-time or on a pre-recorded video.

```
# import cv2 package
import cv2
# import numpy package
import numpy as np
# import keras API package
from keras.models import model_from_json
```

```
# define a dictionary for all classes of emotions
emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: "Neutral",
                5: "Sad", 6: "Surprised"}
```

```
# load json of created model
json_file = open('emotion_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
emotion_model = model_from_json(loaded_model_json)
```

```
# load weights into new model
emotion_model.load_weights("emotion_model.h5")
print("Loaded model from disk")
```

```
# start the webcam feed
cap = cv2.VideoCapture(0)

# pass here the video path
```

```
# cap = cv2.VideoCapture("test_video.mp4")

while True:
    # Find haar cascade to draw bounding box around face
    ret, frame = cap.read()
    frame = cv2.resize(frame, (1280, 720))
    if not ret:
        break
    face_detector = cv2.CascadeClassifier(cv2.data.haarcascades +
                                          "haarcascade_frontalface_default.xml")
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # detect faces available on camera
    num_faces = face_detector.detectMultiScale(gray_frame, scaleFactor=1.3,
                                               minNeighbors=5)

    # take each face available on the camera and Preprocess it
    for (x, y, w, h) in num_faces:
        cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (0, 255, 0), 4)
        roi_gray_frame = gray_frame[y:y + h, x:x + w]
        cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray_frame,
                                                               (48, 48)), -1), 0)

        # predict the emotions
        emotion_prediction = emotion_model.predict(cropped_img)
        maxindex = int(np.argmax(emotion_prediction))
        cv2.putText(frame, emotion_dict[maxindex], (x+5, y-20),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2, cv2.LINE_AA)

    cv2.imshow('Emotion Detection', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```