Convolutional Neural Network based Multiclass Sentiment Analysis to Detect Human Emotions

R.H.Ramawickrama

2020

Convolutional Neural Network based Multiclass Sentiment Analysis to Detect Human Emotions

A dissertation submitted for the Degree of Master of Science in Computer Science

R.H.Ramawickrama
University of Colombo School of Computing
2020

DECLARATION

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: R.H.Ramawickrama

Registration Number: 2017/MCS/063

Index Number: 17440631

____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr. /Ms.

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name:

____

Signature: Date:

Abstract

Sentiment analysis is the area in computer science related to identifying and categorizing opinions implied through a piece of text. Opinions are the most important factor which drives social media, making it vital to understand the underlying emotions of different opinions. This thesis proposes an approach to perform multiclass sentiment analysis using deep convolutional neural networks. The sentiment classes are ‘Happiness’, ‘Sadness’, ‘Anger’, ‘Fear’ and ‘Surprise’. The thesis provides information on the design and the end to end implementation of the convolutional neural network which utilizes one-hot encoding, word embeddings and max pooling to improve the classifier accuracy and performance. It presents how the trained model is evaluated and the results are cross validated with four main performance metrics accuracy, precision, recall and f1-score for each sentiment class. With the proposed approach, the model well predicts four out of the five targeted sentiment classes with an overall model accuracy of 65-70%. Two algorithms are introduced. The first algorithm is to train and test a convolutional neural network which supports multiclass sentiment analysis and the second algorithm is to utilize the trained model for predicting the emotion of a single text and predict the emotions of multiple texts with a consolidated analysis.

Acknowledgements

This thesis and the research behind it would not have been possible without the exceptional support of my supervisor, Dr. G.D.S.P. Wimalaratne, Senior Lecturer at University of Colombo School of Computing (UCSC). His knowledge and exacting attention to detail have been an inspiration and kept my work on track from my first encounter to the final draft of this paper.

My immense gratitude goes to the course coordinator, Dr. L.N.C De Silva for the timely responses for the clarifications and concerns raised and for always thinking from the student's perspective, carrying out everything smoothly from beginning to end.

I am also grateful for the insightful comments offered by the anonymous reviewers and the examiners who evaluated my work during the Project Proposal and Interim Defense presentations. The expertise advice improved this study in innumerable ways and saved me from many errors.

I would also like to express my sincere gratitude to the Post Graduate Division of University of Colombo School of Computing (UCSC) for handling all the submissions and answering the clarifications and concerns raised.

Last but not least, I am most thankful for my fellow colleagues for the support and knowledge shared.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

In the present world, social media has become an essential part in the lives of people and its usage is significantly increasing. According to the Global Digital Report for 2019, 45% of the world's population are now social media users [31]. Social media is used for different purposes such as: to connect with people around the world, to raise awareness on social issues, improve business reputation, for educational purposes and to build communities. Due to these reasons, the use of social media has a drastic increase over the past few years and it continues.

The most important factor which drives social media is the opinions of people and there are many advantages of analysing the opinions. For example, a business could analyse the conversations and customer feedbacks on their products, and take actions accordingly to eliminate poor customer experiences. They can also compare their business outcomes along with their competitors to gain marketing advantages. Educational institutes can identify and address areas of student dissatisfaction based on their conversations. With proper analysis they can also improve on the areas where students are expressing positive opinions. With these advantages, it is vital to understand the underlying emotions of different opinions.

The area in Computer Science related to identifying and categorizing opinions implied through a piece of text is known as sentiment analysis. Over the years, the research on sentiment analysis has been shifted from using classical approaches such as machine learning and lexicon-based to deep learning techniques. Among the deep learning techniques Convolutional Neural Networks (CNN) plays a prominent role due to its significant and promising results. However, the approaches with CNNs have focused on categorizing the emotions as 'Positive' and 'Negative' [16]-[18] and not as primary human emotions such as 'Happiness', 'Sadness' *etc.* Therefore, identifying primary human emotions by utilizing the advantages of CNNs is yet to be explored.

With the timely importance in understanding the underlying emotions of different opinions, the aim of the research is to detect primary human emotions expressed through textual conversations in social media. Based on the research gap identified through the literature review, the objective is to extend the state-of-the-art approach, binary class sentiment classification using CNNs, to support multiclass sentiment classification with enhanced accuracy and efficiency.

A constructive research methodology was followed to conduct the research. The research was scoped around sentence level sentiment analysis (i.e. detecting primary human emotion of a given sentence). The focus was on detecting five primary emotions ‘Happiness’, ‘Sadness’, ‘Anger’, ‘Surprise’ and ‘Fear’. Further categorizations of these emotions were not considered under the study. The domain of the study was social media with the use of Twitter social networking data sets.

The CNN was defined with the use of one-hot encoding, word embeddings and max pooling to improve the classifier accuracy and performance. Based on the proposed solution, the model was able to reach an overall prediction accuracy of 65-70%. Out of all the classes, the model well detects ‘Happiness’ sentiment class. The sentiment class ‘Anger’ has the highest trustability when the model predicts anger as the sentiment for a given text. The four classes ‘Happiness’, ‘Sadness’, ‘Anger’ and ‘Fear’ are perfectly handled by the model and the class ‘Surprise’ is poorly handled by the model.

With the completion of the research a complete algorithm to train and test a CNN which supports multi class sentiment analysis was introduced. Another algorithm was introduced to utilize the trained CNN for predicting the emotion of a single text and predict the emotions of multiple texts with a consolidated analysis. As per the future work, conducting studies on CNNs accuracy and performance in detecting the emotion ‘Disgust’ and further studies on enhancing the accuracy in classifying the emotion ‘Surprise’ were suggested. As the research was conducted with the Twitter data which belongs to the social media domain, it was suggested to extend the same research for sentiment analysis in other domains.

The rest of the thesis is organized as follows. Chapter 2 discusses related work on the area under research with the knowledge gathered from published information in recent years and other credible resources. Chapter 3 provides information on how the solution was constructed based on the limitations identified through the literature review. It also describes the research design, datasets used for training and testing along with the implementation details. Chapter 4 provides details on how a Proof of Concept prototype was developed with the proposed solution. The criteria followed to evaluate the trained CNN model and the results of the evaluation are presented in Chapter 5. Finally, Chapter 6 summarizes and concludes the thesis by highlighting the contributions and suggesting future work.

Chapter 2: Literature Review

This chapter presents background study on the area under research with the knowledge gathered from published information in recent years and information gathered from other credible resources. It first discusses the practical implementations, sub-domains and levels in sentiment analysis. Then it is organized in such a way where it explains the approaches used to perform sentiment analysis, and how research has gradually shifted from machine learning and lexicon-based approaches to deep learning approaches. Furthermore this chapter summarizes the limitations identified and the future work suggested in similar kinds of research.

2.1 Background Study

Sentiment analysis is the area in Computer Science which is related to identifying and categorizing emotions and opinions implied in a piece of text. The process is also known as contextual mining of text which involves systematically identifying, extracting, quantifying, and studying a given text with the use of natural language processing. From marketing to customer service to medicine, Sentiment analysis can be applied to a wide range of areas to analyse the underlying emotions of different opinions expressed and gain valuable insights.

There are multiple technologies built with the use of sentiment analysis such as Tone Analyzer, Receptiviti, Bitext, Synesketch [1] and DeepMoji [24]. Tone Analyzer [25] utilizes IBM Watson [26] to analyze text at document and sentence levels to detect the emotions 'Joy', 'Fear', 'Sadness' and 'Anger' along with analytical, confident and tentative tones expressed through a text. Receptiviti [27] is an analytics platform which is used to analyse the culture of an organization and produce insights to improve the productivity and effectiveness of an organization. It predicts results with the knowledge from the fields of psychology, data science, machine learning and linguistics. Bitext [28] is a Natural Language Processing (NLP) company which develops text analysis tools and NLP middleware to power larger applications in three main areas: Chatbots and Virtual Assistants, Machine Learning Engines with Core NLP services and CX Platforms for sentiment and categorization. Synesketch [29] is an open-source software which recognizes emotions in a text and provides an artistic visualization based on machine learning and lexicon-based approaches. Synesketch algorithms analyze the emotional content of text sentences with the use of three criteria: the type of the emotion, the intensity of the emotion and the valence of the emotion. DeepMoji [24] uses texts on Twitter containing emojis for training a deep learning model that understands how language is used to express emotions. It detects and highlights the emotional impact of words in a given sentence.

As shown by Fig. 1, Sentiment analysis can be classified into several sub-domains such as: Fine-grained Sentiment Analysis, Aspect-based Sentiment Analysis, Intent Analysis and Emotion Detection.

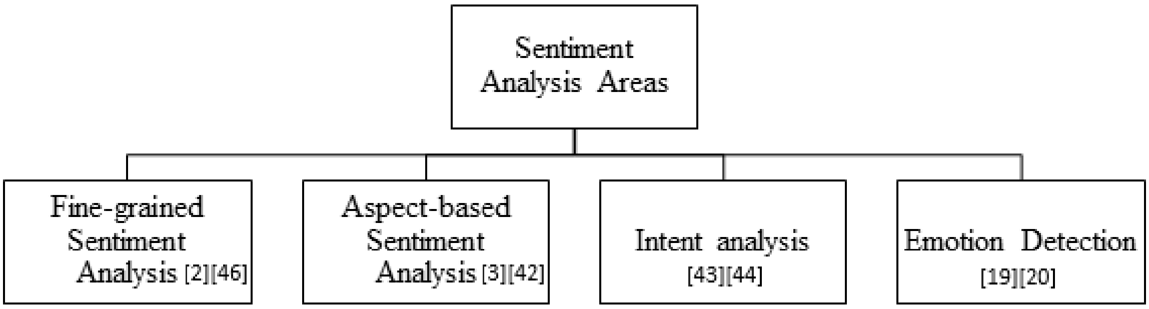

Sentiment Analysis Areas
Fine-grained Sentiment Analysis [2][46]
Aspect-based Sentiment Analysis [3][42]
Intent analysis [43][44]
Emotion Detection [19][20]

Fig. 1. Sentiment analysis sub-domains

Fine-grained sentiment analysis focuses on the polarity of opinions [2]. With this, opinions categorized as 'Positive', 'Negative' and 'Neutral' can be further drilled down as 'Very Positive', 'Positive', 'Neutral', 'Negative' and 'Very Negative'. Aspect-based sentiment analysis is about identifying which aspects, features of a product people are mostly talking about [3]. Intent analysis, as the name suggests, focuses on the intention behind the text. It identifies a complaint, a question, a request made through a text. Emotion detection focuses on categorizing opinions into primary human emotions such as 'Happiness', 'Sadness', 'Fear', 'Disgust', 'Anger' and 'Surprise'.

Fig. 2 depicts the different levels of granularities or scopes in which we can conduct Sentiment Analysis.

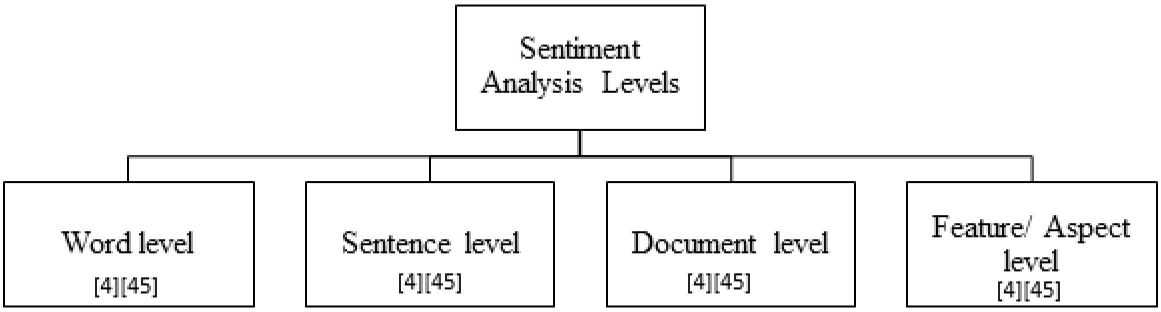


Fig. 2. Sentiment analysis levels

Word Level focuses on finding the sentiment of a given word. Sentence level predicts the sentiment of a given sentence. Document level provides the overall sentiment of a document. For example, the whole document is classified either as 'Positive' or 'Negative'. Aspect or feature level sentiment classification identifies features of a given product and assigns each feature with a sentiment.

The classical approaches for performing sentiment analysis can be categorized into two main areas as machine learning and lexicon-based [4] as shown by Fig. 3.

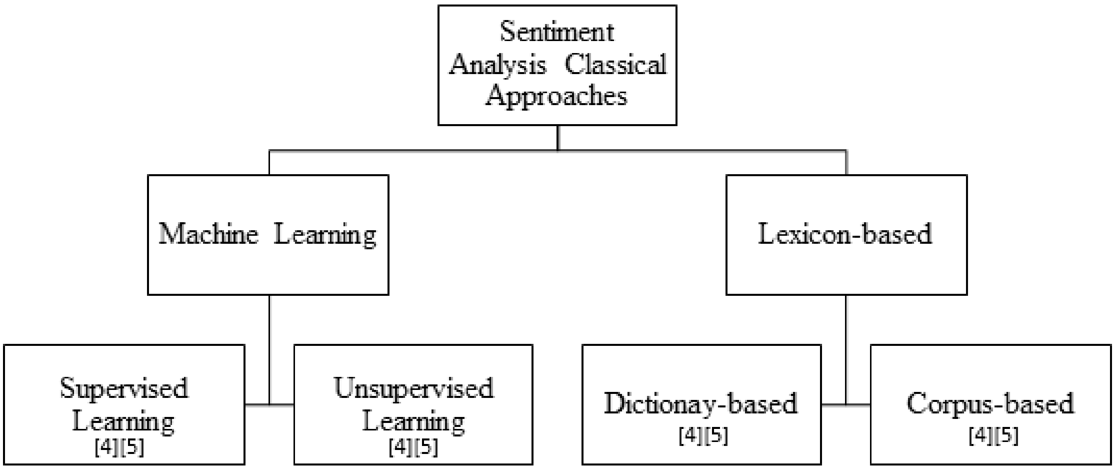

Sentiment Analysis Classical Approaches
Machine Learning
Lexicon-based
Supervised Learning [4][5]
Unsupervised Learning [4][5]
Dictionay-based [4][5]
Corpus-based [4][5]

Fig. 3. Sentiment analysis classical approaches

Machine learning is the process of making a computer system to perform a specific task based on the patterns identified and through inferences without following a set of pre-defined instructions. Machine learning techniques can be classified as supervised and unsupervised learning. Lexicon-based approach adopts a lexicon (i.e. the complete set of meaningful units in a language) to perform sentiment analysis. First, it cross verifies the words of a given sentence with the words in the lexicon where the sentiment class is pre-tagged, and tag the appropriate sentiment for the word in the sentence. Then, the sentiment analysis is performed by counting and weighing the tagged sentiment classes. Lexicon-based techniques can be further classified as dictionary and corpus based approaches.

Experimental results show that both machine learning and lexicon-based approaches have their own advantages and disadvantages [5]. For example, with the use of machine learning approaches, task specific pre-trained models can be created with the ability to adapt. But, for such training, it needs to access labelled data which could be costly or scarce. Also with the increase of text diversification, machine learning needs more processing which results in reduced efficiency. Lexicon-based approaches can reflect the characteristics of unstructured text data but it excessively relies on emotional dictionaries. When there is no definition of a certain word, the accuracy decreases. Due to these advantages and disadvantages, separate studies have been conducted to have hybrid approaches for sentiment analysis by using the advantages of each approach to overcome disadvantages [6]-[11].

Discussed below is the critical review on the research focused on utilizing machine learning and lexicon-based approaches to detect primary human emotions.

Chatterjee et al. [6] have proposed an approach to classify the emotion of user utterance into four output classes ‘Happiness’, ‘Sadness’, and ‘Anger’ and ‘Others’. For this they have combined sentiment analysis and semantic analysis both together and introduced a model named ‘Sentiment and Semantic Based Emotion Detector (SS-BED)’. Semantic analysis understands data under various logical clusters without limiting to the preset categories ‘Positive’ and ‘Negative’. A given text may consist of both a negative word and a positive word, but the overall emotion of the text can be negative. For such scenarios with the help of semantic analysis, emotion can be identified accurately. As per their proposed model, the user utterance was given as the input for two layered neural network where one layer was based on a semantic word embedding and the other based on a sentiment word embedding. The learned semantic and sentiment feature representation from these two layers were then passed to a fully connected network where the features were analysed and the probabilities for each emotion class were generated. As for future work, the researchers have planned to extend the same approach to support the detection of the rest of the emotional classes such as ‘Surprise’, ‘Fear’ and ‘Disgust’.

In a study conducted by S. Zhu et al. [12], the researchers have proposed an approach for emotion classification by combining two corpuses together as a solution for the data deficiency problem. With the approach, the annotated data from one corpus can be used for the emotion classification on another corpus. The classification results were refined with an Integer Linear Programming (ILP) optimization approach. The experimental results have demonstrated that

ILP optimization improves the performance. However, annotation guidelines being different between the two emotion corpora was one of the major challenges they have faced. For example, one corpus categorizes emotions into seven classes where each instance includes both primary emotion and secondary emotion, or just has one primary emotion. The other corpus includes four emotion classes along with complex emotions for each instance as positive, neutral and negative. As per their future work they have suggested modifications on the ILP for further improvement and adapting the proposed approach to other NLP tasks where two or more corpora are available.

X. Zhang et al. [13] have researched on identifying the emotion distribution of a text having multiple emotions with different intensities by learning a mapping function. The identified relations between the emotions were incorporated into the learning algorithm to enhance the accuracy. Experimental results have shown that the proposed approach performs better than both the state-of-the-art methods for emotion detection and multi-label learning methods. As per future work, they have suggested investigating the efficiency of the proposed approach with the use of different datasets, exploring the other possibilities in capturing the inter-relations of emotions.

As a result of the research conducted by M. Larsen et al. [14], the group has developed a system named ‘We Feel’ to analyse variations in emotional expression regionally and globally. The results from the analysis were validated against the known patterns of variation in mood, and reported. They have used automatically annotated Twitter data as their source for analysis. For this, an emotional vocabulary was created by collecting emotional terms from multiple sources and were organized with the use of crowdsourcing**.** They have classified each word into six primary emotion categories of ‘Love’, ‘Joy’, ‘Surprise’, ‘Anger’, ‘Sadness’ and ‘Fear’. They have also added extended support for 25 subgroups of secondary emotions.

Even though the classical methods such as machine learning and lexicons for sentiment analysis have shown promising results, gradually the research focus has shifted towards achieving better accuracy and efficiency using deep learning methods. Deep learning is an extended version of Artificial Neural Networks (ANN), where ANN is a sub division of Machine Learning.

Deep learning can be categorized into two areas as supervised and unsupervised. Supervised methods include: Artificial Neural Networks, Convolutional Neural Networks and Recurrent

Neural Networks. Unsupervised methods include: Self-Organizing maps, Deep Boltzman Machines and AutoEncoders [30]. The deep learning techniques for sentiment analysis includes Convolutional Neural Networks, Recursive Neural Networks, Deep Neural networks, Recurrent Neural networks and Deep Belief Networks [15].

Discussed below is the critical review on the research focused on using deep learning techniques for sentiment analysis.

S. Rani et al. [16] have conducted a study on deep learning based sentiment analysis using CNNs with the use of a set of collected Hindi movie reviews. The sentiment analysis has been done by changing different configuration settings of CNN and experimenting on them. The model they have proposed was able to achieve better performance than traditional machine learning approaches and with an accuracy of 95%. As per future work, they have suggested extending the experimentation for other deep learning methods by using datasets for different domains such as social media, politics *etc.*

On a study conducted by Z. Jianqiang et al. [17], the researchers have considered a deep CNN for Twitter sentiment analysis. They have researched on improving the accuracy and analysis speed with the use of a convolution algorithm to train the deep neural network. The steps of their approach includes: performing unsupervised learning with the use of Twitter corpora, learning global vectors for word representation, concatenating the word representations with the knowledge on polarity score features and state-of-the-art features and produce feature sets, combining the feature sets and feeding into a deep CNN to train and predict the sentiment of tweets. Based on their findings, they have identified multiple advantages of CNNs such as: reducing the problems faced with scarcity of training data, giving the priority for the important features in a tweet, selecting the most important features in the tweet effectively, effectively catching the context sentiment information from the tweet and directly modelling them**,** retaining the word order information and improving the classification performance.

On a study conducted by N. Nedjah et al. [18], the researchers have proposed an approach for sentiment classification using CNN by studying the impact of the hyper-parameters on the performance of the model. For this they have considered different word embeddings. Word embeddings are used to translate text into vector format supported by CNNs as they cannot directly accept the sentences as they are. The steps in their approach includes: pre-processing and transforming the input text into a list of words, transforming each word in the list into a vector having fixed dimensions and then finally classifying the text. The evaluation results indicate that there is a major impact on the performance based on the hyper-parameter configuration and the evaluated data set used.

N. Majumder et al. [19], have proposed an approach for emotion detection in conversations using Recurrent Neural Networks (RNN). Their study was focused under the assumption that three major aspects: the speaker, the context from the preceding utterances, and the emotion of the preceding utterances are affecting the emotion of conversation. The proposed system consists of three Gated Recurrent Units (GRU) to model the aspects. The incoming utterances were fed into two GRUs. With this, the information of all preceding utterances by different parties in the conversation were gathered. Then the updated speaker state was input into the final GRU to detect the emotion of a given utterance. Their model has outperformed the state-of-the-art approaches on two distinct datasets in both textual and multimodal settings. As per their future work, they have planned to support their model for conversations with more than two speakers.

S. Smetanin [20] has proposed a Bidirectional Long Short-Term Memory (LSTM) network for contextual emotion detection in textual conversations. The proposed architecture consists of two bidirectional LSTM units to learn semantic and sentiment feature representations and capture user-specific conversation features in order to enhance the accuracy. The two LSTMs were designed to separately analyse the utterances of the two users in the conversation. As per the future directions, they have suggested advanced usage of techniques to handle imbalanced data and to consider the application of character level language models.

Although sentiment analysis focuses on detection of emotions from text, significant research has been conducted on emotion detection from audio using the deep learning techniques discussed previously. Given below is the critical review on such approaches in order to identify feasibility of them to be applied for emotion detection from text.

D. Bertero et al. [22] have proposed an approach to detect emotions of real-time speech using CNNs. Their model has been trained from raw audio data to support the three emotion classes ‘Happiness’, ‘Sadness’ and ‘Anger’. Their approach was able to reach an average accuracy of 66.1% with an evaluation time of a few hundred milliseconds. For future directions, they have suggested extending the analysis to support for other emotions classes and other data domains with more speech data.

On a study conducted by S. Lalitha et al. [21], an investigation has been carried out to identify the effect of various perceptual features for emotion detection using Deep Neural Networks (DNN) as these features contain valuable information about the speaker. Their experimental results have proved that the perceptual features like Revised Perceptual Linear Prediction coefficient (RPLP), Bark Frequency Cepstral Coefficients (BFCC) and Inverted Mel Frequency Cepstral Coefficients (IMFCC) have a less importance in detecting the emotions when compared with the rest of the features. Their DNN having three hidden layers were able to achieve a better performance in classifying seven emotions. As per future work, they have suggested working on multi-corpus acted and natural databases having perceptual speech features, investigating the performance of other deep models in discriminating emotions with the use of perceptual features, exploring multimodal emotion detection, and finding a solution for imbalanced datasets in speech emotion corpora.

S. Yoon et al. [23] have researched on speech emotion detection using a multi-hop attention mechanism utilizing both acoustic and textual information. Their approach first identified the relevant textual data corresponding to the audio signal. Then the textual data was applied to attend parts of the audio signal. The outputs were then fused for classification. The proposed framework uses two bi-directional LSTMs to identify the hidden representations of the utterance and to automatically produce the correlations. Their approach has outperformed the state-of-the-art system by 6.5%.

Table I summarizes the research on emotion detection carried with the use of deep learning techniques.

TABLE I
Summary on the research on emotion detection using deep learning techniques

Research	Deep Learning Technique	Sentiment Classes	Emotion Detection Source (Text/ Audio)
"Deep Learning Based Sentiment Analysis Using Convolution Neural Network", S. Rani et al., 2018 [16]	Convolutional Neural Networks (CNN)	'Positive' and 'Negative'	Text (Sentiment Analysis)

“Deep Convolution Neural Networks for Twitter Sentiment Analysis”, Z. Jianqiang et al., 2017 [17]	Convolutional Neural Networks (CNN)	‘Positive’ and ‘Negative’	Text (Sentiment Analysis)
“Sentiment analysis using convolutional neural network via word embeddings”, N. Nedjah et al., 2019 [18]	Convolutional Neural Networks (CNN)	‘Positive’ and ‘Negative’	Text (Sentiment Analysis)
“A first look into a Convolutional Neural Network for speech emotion detection”, D. Bertero et al., 2017 [22]	Convolutional Neural Networks (CNN)	‘Happy’, ‘Sad’, ‘Angry’	Audio
“DialogueRNN: An Attentive RNN for Emotion Detection in Conversations”, N. Majumder, 2019 [19]	Recurrent Neural Networks (RNN)	‘Happy’, ‘Sad’, ‘Neutral’, ‘Angry’, ‘Excited’, ‘Frustrated’	Text (Sentiment Analysis)
“Proceedings of the 13th International Workshop on Semantic Evaluation”, S. Smetanin, 2019 [20]	Recurrent Neural Networks (RNN)	‘Happy’, ‘Sad”, ‘Angry’	Text (Sentiment Analysis)
“Enhanced speech emotion detection using deep neural networks”, S. Lalitha et al., 2018 [21]	Deep Neural Networks (DNN)	‘Fear’, ‘Anger’, ‘Boredom’, ‘Disgust’, ‘Happy’, ‘Neutral’, ‘Sad’	Audio
“Speech Emotion Recognition Using Multi-hop Attention Mechanism”, S. Yoon et al., 2019 [23]	Deep Neural Networks (DNN)	‘Happy’, ‘Sad”, ‘Angry’	Audio

Among the deep learning approaches for sentiment analysis, CNNs play a prominent role [16]-[18]. Since these approaches have focused on detecting emotions as ‘Positive’ and ‘Negative’, further studies should be conducted on identifying primary emotions using CNN due to its significant experimental results from previous research. However, detecting primary emotions with the use of CNN has been done for audio data [22] in order to detect three primary human emotions. Apart from CNN, Recurrent Neural Networks (RNN) [19] [20] has been used for primary emotion detection from text. Deep Neural Networks (DNN) has been used for primary emotion detection from audio data [21][23].

2.2 Summary

This chapter summarized the previous studies on sentiment analysis along with its practical implementations, sub-domains and different levels. It explained how the research on Sentiment Analysis was previously focused on machine learning and lexicon-based approaches and how researchers studied hybrid approaches to enhance accuracy and performance. Then the chapter discussed how the research gradually shifted from machine learning and lexicon-based approaches to deep learning approaches. A detailed analysis on deep learning approaches were then presented. Furthermore this chapter summarized the limitations identified and the future work suggested in similar kinds of research. Based on the literature review conducted it was identified that even though a significant amount of research has been done on binary class sentiment analysis using CNNs with promising results, there is a gap in performing multiclass sentiment analysis using the same. Therefore further studies should be conducted to bridge the gap by extending the binary class classification to support multiclass classification, which could be used for emotion detection.

Chapter 3: Methodology

This chapter outlines the research methodology that was followed in the study. It provides information on how the solution was constructed based on the limitations identified through the literature review. The chapter describes the research design and the reasons for the choices. The datasets that were used for training and testing are also described and the implementation details are included.

A constructive research methodology was followed when conducting the research. It is a systematic approach where the importance is given to both practical and theoretical aspects of an idea with research potential. The solution is constructed based on both practical and theoretical knowledge gathered. In this research methodology, theoretical demonstration and practical implementation are considered as valid outcomes of the research process. Fig. 4 depicts the phases in the constructive research methodology.

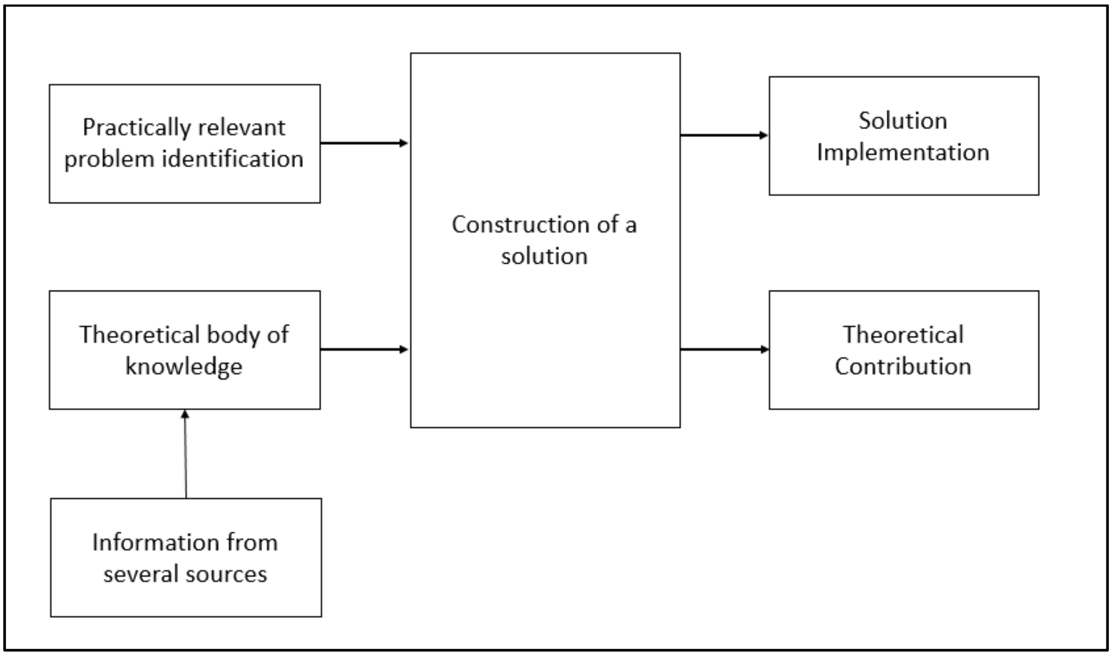

Practically relevant problem identification
Theoretical body of knowledge
Information from several sources
Construction of a solution
Solution Implementation
Theoretical Contribution

Fig. 4. Constructive research based approach used in the thesis

As explained in Chapter 1, the most important factor which drives social media is the opinions of people. This is a very valuable asset since if analysed properly we can derive valuable information out of them. Therefore, it is vital to have a system which can support it and provide a detailed analysis on the opinions by converting data into valuable information.

As discussed in the literature review in Chapter 2, among the deep learning techniques used for sentiment analysis, CNNs play a prominent role [16]-[18]. Since previous studies have focused on detecting emotions as ‘Positive’ and ‘Negative’, it was identified that further studies should be conducted on identifying primary emotions using CNNs due to its significant experimental results from previous research.

3.1 Research Design

Fig. 5 depicts the high level design of the proposed approach to perform multiclass sentiment analysis using CNN.

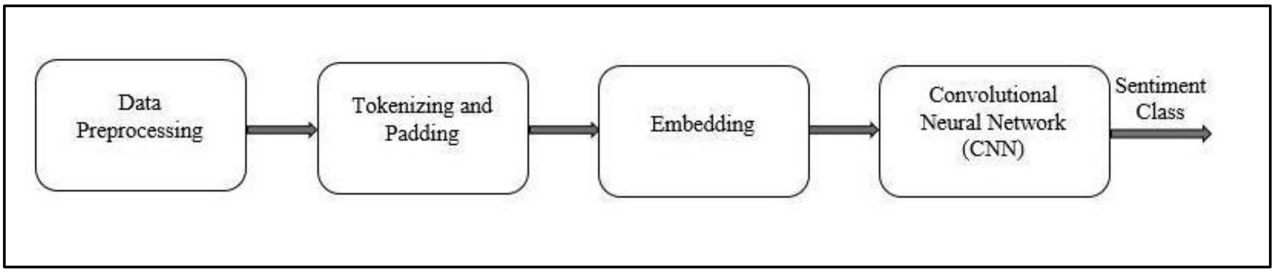


Fig. 5. High level research design

Detailed descriptions on each module of the high level design is described below.

3.1.1 Data pre-processing

The data pre-processing step had two major steps: encoding the sentiment labels and pre-processing the phrases. Encoding was needed to convert the sentiment labels, which are categorical, into a numerical form that could be provided into the deep learning model to do a better job in prediction. Two methods were identified to convert a categorical value into a numerical format.

- Integer Encoding [37]: Integer encoding, which is also known as label encoding is where an integer value is assigned for each unique category. For example, ‘Happiness’ is 1, ‘Sadness’ is 2 and ‘Fear’ is 3. But the disadvantage with this encoding is that it assumes that a certain category is better when it has a higher value assigned.
- One-Hot Encoding [37]: In one-hot encoding, each unique class is represented using a binary value. ‘1’ is given for the respective class and ‘0’ is given for other classes.

From the above two approaches, for sentiment label encoding, one-hot Encoding was selected. Fig. 6 displays how each sentiment class was represented using one-hot encoding.

Happiness	Sadness	Anger	Surprise	Fear

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Fig. 6. One-Hot encoding representation of sentiment classes

Following were the pre-processing steps to clean the phrases.

- Splitting sentences / phrases on white space to get words
- Removing all punctuation marks from words
- Removing all words that are not purely comprised of alphabetical characters
- Lower casing the data
- Removing all the stop words (Stop words are a set of commonly used words in any language. For example, in English, ‘the’, ‘is’, ‘and’ can be considered as stop words)

3.1.2 Tokenizing and Padding

Text data must be encoded as numbers to be used as input for deep learning models since we cannot feed raw text directly into them. This process is known as Tokenizing where an integer is assigned to each word. The sentences / phrases are defined as a pad sequence to have the same length for all the sentences. The example given below describes how the phrase “Tokenize and Pad sequence” is tokenized and padded.

Phrase: ‘Tokenize and Pad sequence’
Let ‘tokenize’ = 1, ‘and’ = 2, ‘pad’ = 3 and ‘sequence’ = 4
Then, ‘Tokenize and Pad sequence’ = [1,2,3,4]
If max sequence length is 10, then after padding,
‘Tokenize and Pad sequence’ = [0,0,0,0,0,0,1,2,3,4]

3.1.3 Embedding

An embedding is a mapping of a discrete or categorical variable to a vector of continuous numbers [32]. Embedding can be used to learn the contextual information of words thus improving the accuracy which also greatly improves the efficiency of a deep learning network. Fig. 7 displays how a given phrase is arranged after applying embedding.

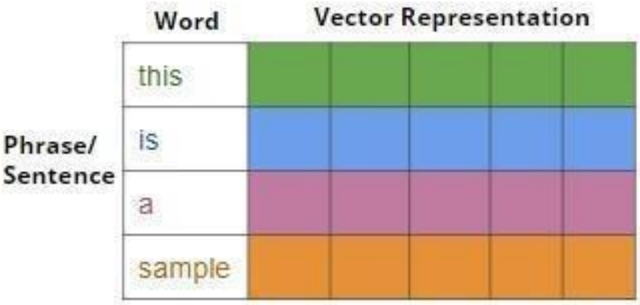


Fig. 7. Word embedding representation

3.1.4 Convolutional Neural Network

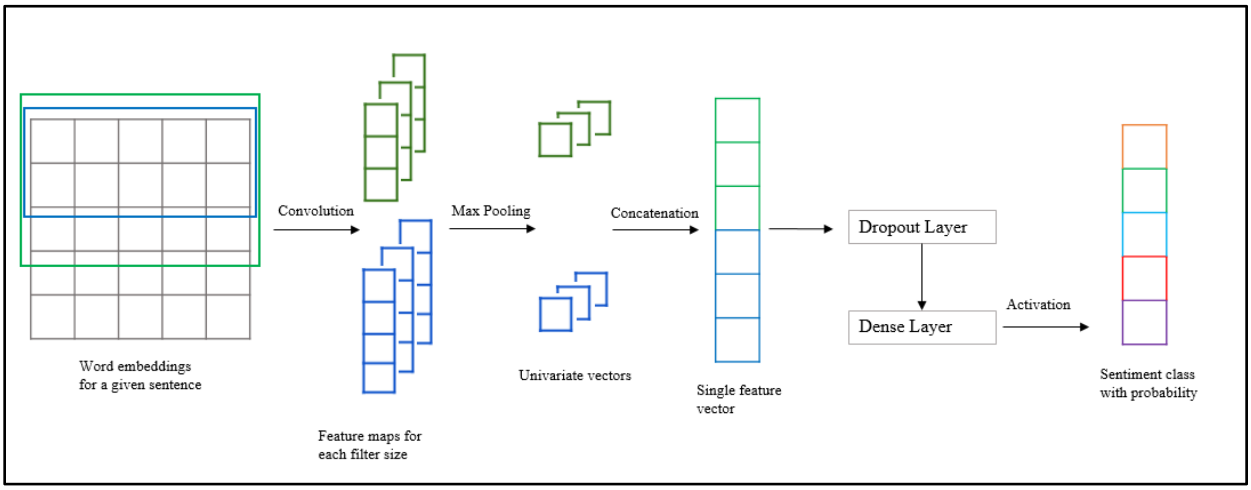


Fig. 8. Convolutional neural network design

Described below is the end to end process in the CNN displayed in Fig. 8 along with the selection of the hyper parameters. Hyper parameters are the variables which determine the network structure and how the network is trained.

The word embedding of a given text was provided as the input to the CNN. First, sliding windows with heights were defined. These are also known as filter sizes or kernel sizes.

It determines how many words are considered when training the feature detector. For example, as per the given figure, the heights of the sliding windows are 2 (blue) and 3 (green). The length of the sliding window covers the whole word. Each filter size is associated with a convolution layer which runs the sliding window on the provided text and generates feature maps. Based on the length of the phrases in the train data, 5 filter sizes were decided starting from 2 as 2, 3, 4, 5 and 6, as using smaller filter sizes is computationally efficient.

Next, the feature maps generated by each convolutional layer were reduced to univariate vectors by applying max pooling. Max pooling is a sample-based discretization process and it is used to reduce the dimensionality of an input representation. By reducing the dimensionality, assumptions can be made easily about features contained in the sub-regions. It prevents overfitting by taking the max value of the learned features and also reduces the number of parameters to learn which in turn reduces the computational cost. To perform max pooling, again the sliding window approach was used.

Finally, the univariate vectors from each max pooling layer were concatenated together to form a single feature vector. On this feature vector, a dropout layer and a dense layer was applied.

A dropout layer offers a very computationally cheap and remarkably effective regularization method to reduce overfitting and improve generalization in deep neural networks. Deep learning neural networks are likely to quickly overfit on a training dataset with few examples. By applying a dropout layer, the network becomes less sensitive to react to smaller variations in the data and further increase accuracy on unseen data.

Finally, an activation function was applied to the dense layer in order to produce a probability distribution over the output sentiment classes. Activation functions are mathematical equations which determine the output of a neural network. The function is attached to each neuron in the network, and determines whether the neuron should be activated or not, based on whether each neuron’s input is relevant for the model’s prediction. There are different types of activation functions used in neural networks and each of them has their own advantages and disadvantages. As the research under study focuses on multi class classification, softmax activation function was used due to its ability to handle multiple classes. This function normalizes the outputs for each class between 0 and 1, and divides by their sum, giving the probability of the input value being in a specific class.

3.2 Datasets for training and testing

For the model training and evaluating purposes, two sets of twitter data were identified where they have the tweet along with the sentiment label.

- Dataset 1: ‘Sentiment Analysis in Text dataset’ from crowdflower.com. A subset of this data has been used in an experiment uploaded to Microsoft's Cortana Intelligence Gallery.
- Dataset 2: 8th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis (WASSA-2017) Shared Task on Emotion Intensity (EmoInt).

The distribution of the sentiment labels in each dataset is consolidated in Table II.

TABLE II
Sentiment label distribution in the identified datasets

Sentiment	Presence of Sentiment Label	
	Dataset 1	Dataset 2
Happiness	Yes	Yes
Sadness	Yes	Yes
Fear	No	Yes
Anger	Yes	Yes
Surprise	Yes	No

In order to have a complete dataset which supports training and testing for all five primary emotions ‘Happiness’, ‘Sadness’, ‘Anger’, ‘Surprise’ and ‘Fear’, the two datasets were combined into one single dataset. Fig. 9 displays the distribution of the final data set after concatenating.

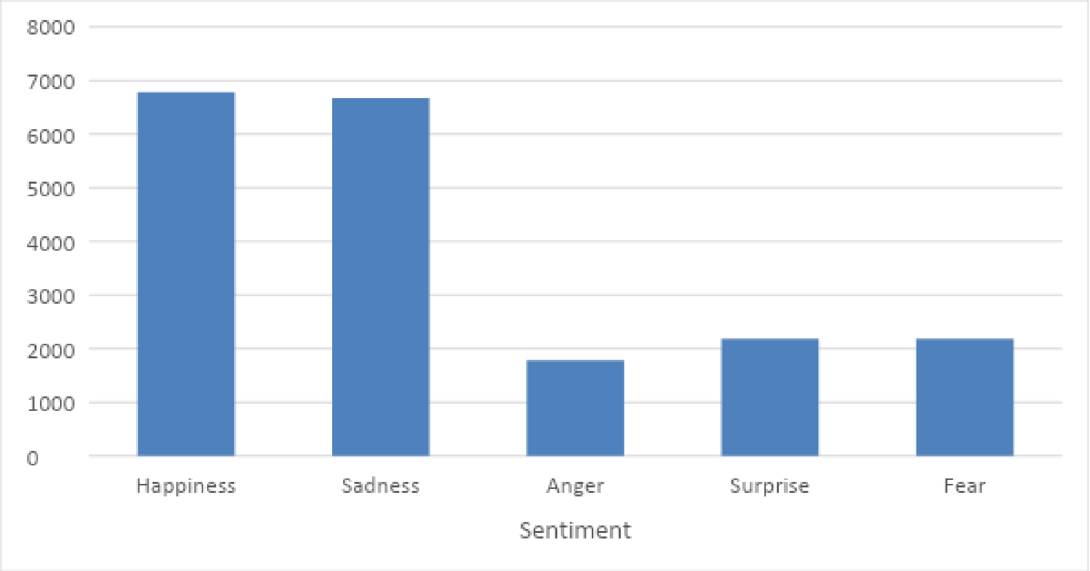


Fig. 9. Final data set distribution after concatenating

3.3 Implementation

The sections below describe how the end to end flow of the CNN was implemented.

3.3.1 Technology Stack

The CNN model training and testing was carried out in Windows 10 environment with the below described technology stack

- Anaconda [33]: Distribution platform for Python language and it enables easy installation of Python modules in a machine.
- Tensorflow [34]: Open-source artificial intelligence library which can be used to implement multi layer deep learning models directly or by using wrapper libraries.
- Keras [35]: Open-source neural-network library written in Python which can be run on top of TensorFlow. Its main advantages include being user-friendly, modular, and extensible and also enables fast experimentation with deep neural networks.
- Spyder [36]: Open source cross-platform integrated development environment built for scientific programming in Python language.

3.3.2 Defining the CNN

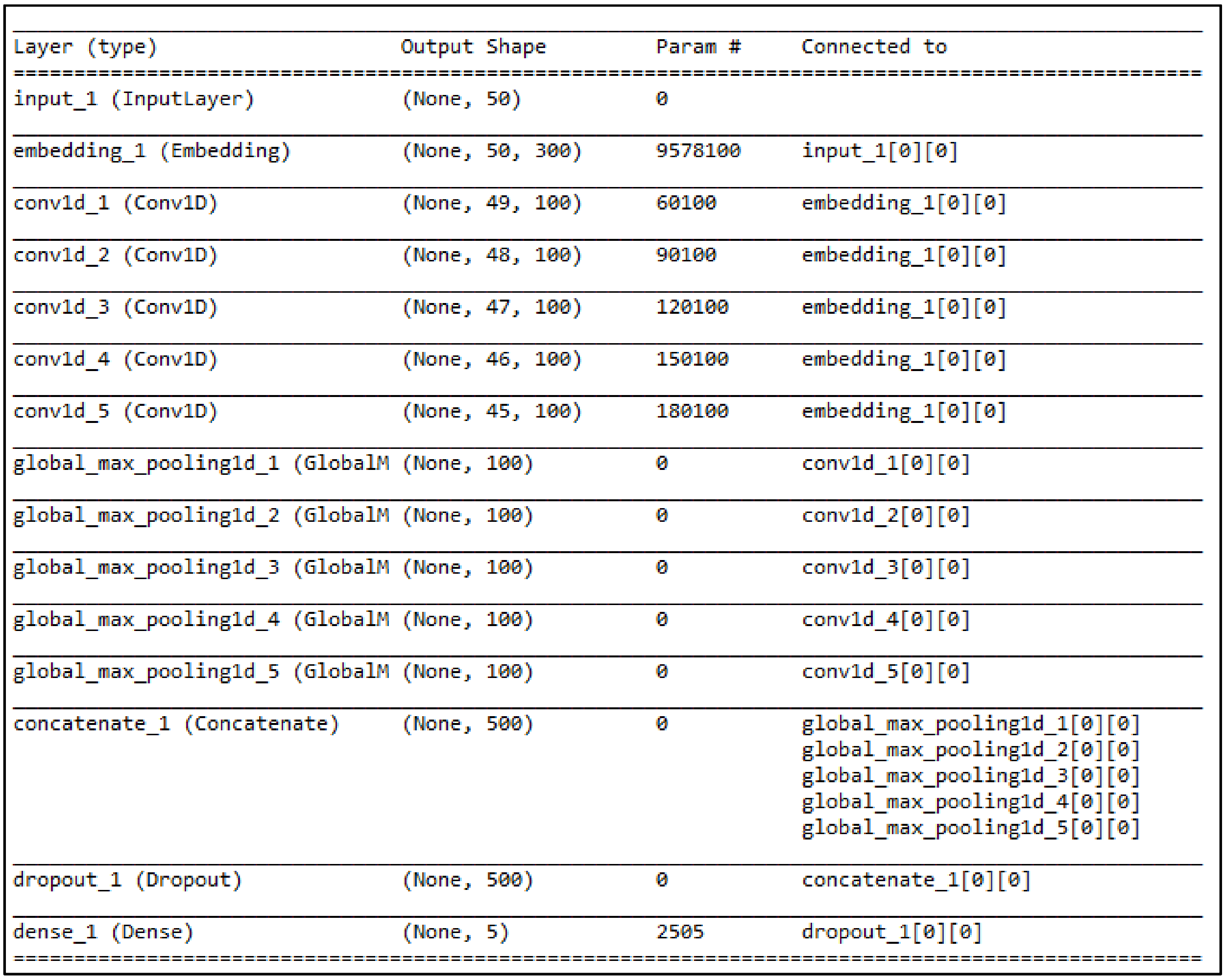
First, the data set was read as a CSV (Comma-separated values) file. Next, the sentiment labels which were in categorical format were converted into a format which could be provided into the deep learning model, for a better prediction. For this one-hot encoding was used.

The next step was data cleaning. For this, first the punctuation marks were removed. Next, each phrase was tokenized by using the word_tokenize method in Python's Natural Language TookKit (NLTK) library [38]. Then the data was set to lowercase and the stop words were removed using NLTK’s stopwords so that the model can focus on the important words instead.

After cleaning the data, the dataset was split into two parts as train and test. 90 % of the data was allocated for training and 10 % for testing. Then a training vocabulary was built with the use of train data.

Next, an integer was assigned to each word. As all the training sentences must have the same length or input shape, the sentences were padded. Then the embeddings from Google News Word2Vec model were taken and saved corresponding to the sequence number assigned to each word. A random vector was saved for the words which did not have a corresponding embedding.

As the next step, CNN was implemented. It was defined in such a way that text as a sequence was taken as the input. The embeddings matrix was passed to the embedding layer. Five different filters with sizes 2, 3 ,4 ,5 and 6 were applied to each phrase, and max pooling layers were applied to each layer. All the outputs were then concatenated. For the concatenated feature vector a dropout layer and a dense layer was applied. Fig. 10 displays the output from the Python console which depicts the layers of the defined CNN and how each layer is connected.



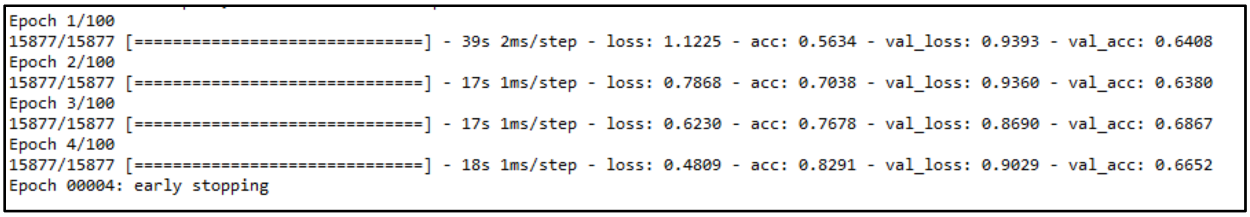
```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 50)           0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 50, 300)      9578100     input_1[0][0]
__________________________________________________________________________________________________
conv1d_1 (Conv1D)               (None, 49, 100)      60100       embedding_1[0][0]
__________________________________________________________________________________________________
conv1d_2 (Conv1D)               (None, 48, 100)      90100       embedding_1[0][0]
__________________________________________________________________________________________________
conv1d_3 (Conv1D)               (None, 47, 100)      120100      embedding_1[0][0]
__________________________________________________________________________________________________
conv1d_4 (Conv1D)               (None, 46, 100)      150100      embedding_1[0][0]
__________________________________________________________________________________________________
conv1d_5 (Conv1D)               (None, 45, 100)      180100      embedding_1[0][0]
__________________________________________________________________________________________________
global_max_pooling1d_1 (GlobalM (None, 100)          0           conv1d_1[0][0]
__________________________________________________________________________________________________
global_max_pooling1d_2 (GlobalM (None, 100)          0           conv1d_2[0][0]
__________________________________________________________________________________________________
global_max_pooling1d_3 (GlobalM (None, 100)          0           conv1d_3[0][0]
__________________________________________________________________________________________________
global_max_pooling1d_4 (GlobalM (None, 100)          0           conv1d_4[0][0]
__________________________________________________________________________________________________
global_max_pooling1d_5 (GlobalM (None, 100)          0           conv1d_5[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 500)          0           global_max_pooling1d_1[0][0]
                                                                 global_max_pooling1d_2[0][0]
                                                                 global_max_pooling1d_3[0][0]
                                                                 global_max_pooling1d_4[0][0]
                                                                 global_max_pooling1d_5[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 500)          0           concatenate_1[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 5)            2505        dropout_1[0][0]
==================================================================================================
```

Fig. 10. Layers of the convolutional neural network

3.3.3 Training the CNN

The final step was training CNN. For this, the 'batch size' was set to 32 which is the standard size for CNNs. Batch size is the amount of data which the model will consider at each round. 'Number of epochs' was set to a higher number which is 100. The number of epochs is the number of times the model will loop around and learn. Keras can separate a portion of the training data into a validation dataset and evaluate the performance of the model on that validation dataset for each epoch. When training the model, each epoch displays the loss and accuracy on both the training dataset and the validation dataset. The validation split was set to 0.1 so that 10% of the training data will be used for validation when training the model.

With the increasing number of epochs, the model may lead to overfitting. To avoid this, the concept of early stopping was used. With early stopping, we can specify an arbitrary large number of training epochs and stop training once the model performance stops improving on the validation dataset. Keras supports early stopping via a callback called 'EarlyStopping'. It allows you to specify the performance measure to monitor and the trigger, and once triggered, it will stop the training process. Validation loss was set as the performance measure for early stopping with minimum loss as the target. Fig 11 displays how the training stopped with 4 epochs with the support of early stopping.



```
Epoch 1/100
15877/15877 [==============================] - 39s 2ms/step - loss: 1.1225 - acc: 0.5634 - val_loss: 0.9393 - val_acc: 0.6408
Epoch 2/100
15877/15877 [==============================] - 17s 1ms/step - loss: 0.7868 - acc: 0.7038 - val_loss: 0.9360 - val_acc: 0.6380
Epoch 3/100
15877/15877 [==============================] - 17s 1ms/step - loss: 0.6230 - acc: 0.7678 - val_loss: 0.8690 - val_acc: 0.6867
Epoch 4/100
15877/15877 [==============================] - 18s 1ms/step - loss: 0.4809 - acc: 0.8291 - val_loss: 0.9029 - val_acc: 0.6652
Epoch 00004: early stopping
```

Fig. 11. CNN training output with early stopping

3.4 Summary

This chapter outlined how the solution was created based on the limitations identified through the literature review. It first presented the high-level research design and each step in the design was explained in detail along with reasoning for each parameter selection. Next, the datasets used for training and testing the model was presented with a clear picture on data distributions. Finally, the chapter described how the solution was implemented by explaining the technology stack and the development steps. The model training outcome was described at the end along with the reasoning for training parameter selections.

Chapter 4: Proof of Concept Prototype

This chapter provides details on the system which was developed as a proof of concept prototype to display CNNs ability to predict emotions. It explains how the system was implemented to support single and multiple emotion prediction.

The developed prototype has the following two major functionalities:

- Submitting a text and predicting the emotion of the text along with confidence levels
- Submitting multiple texts and predicting the overall emotion count for the texts

4.1 Predicting the emotion of a single phrase

As shown by Fig. 12, when a phrase is submitted from the user interface, it is passed to the Java Servlet layer. The Java layer invokes the Python program by passing the submitted phrase as an input parameter. The Python program which is written on top of the trained CNN, has the ability to predict the emotion of a given text.

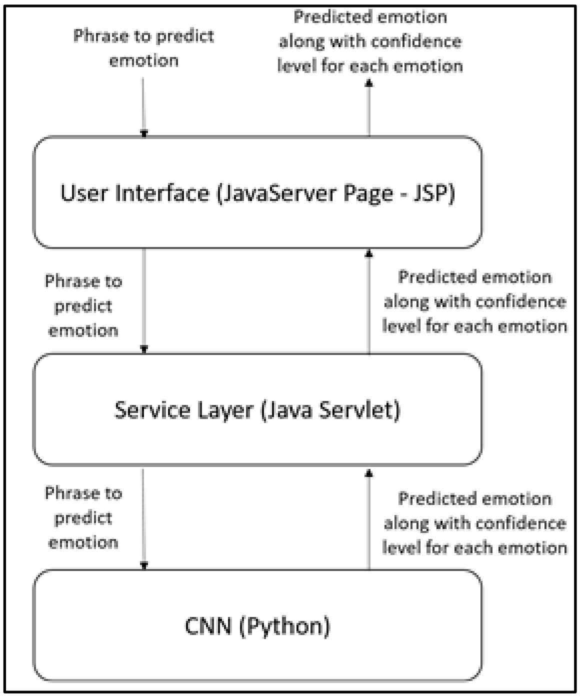
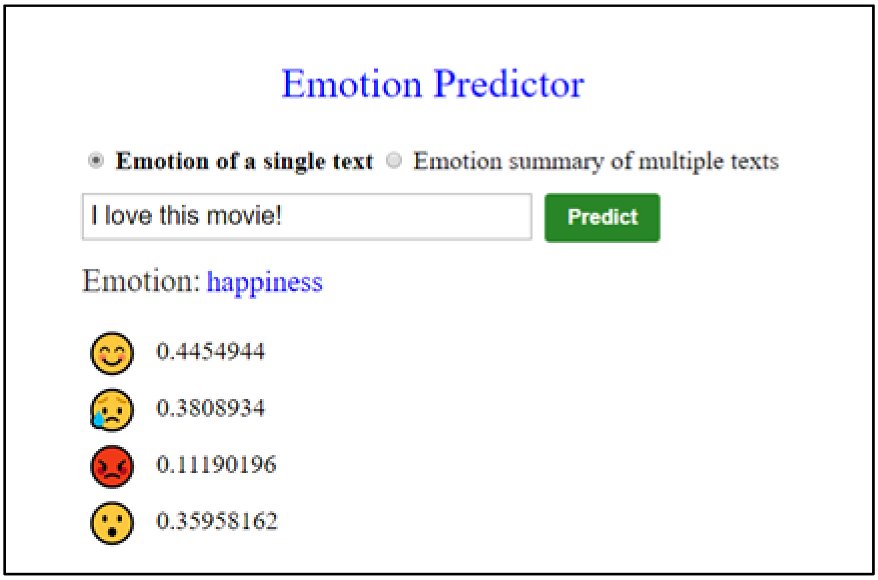

Phrase to predict emotion
Predicted emotion along with confidence level for each emotion
User Interface (JavaServer Page - JSP)
Phrase to predict emotion
Predicted emotion along with confidence level for each emotion
Service Layer (Java Servlet)
Phrase to predict emotion
Predicted emotion along with confidence level for each emotion
CNN (Python)

Fig. 12. POC prototype - single prediction flow

The final output is displayed to the user as shown by Fig. 13.



Emotion Predictor

● **Emotion of a single text** ○ Emotion summary of multiple texts

I love this movie!

Emotion: happiness

😊 0.4454944

😢 0.3808934

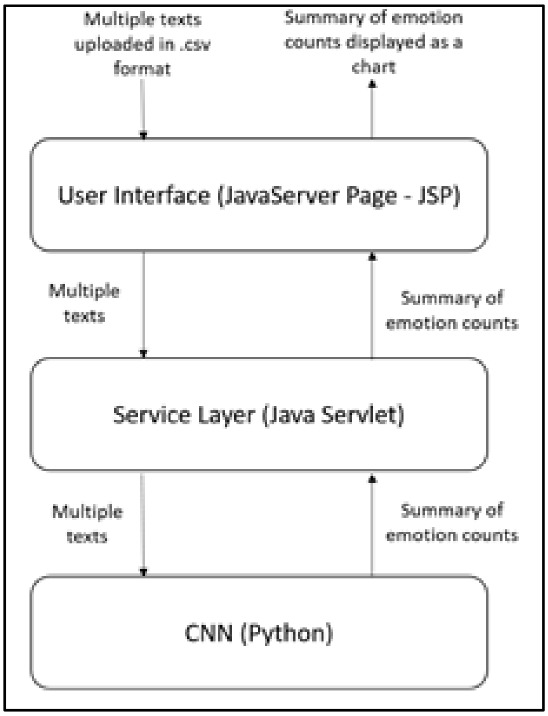
😡 0.11190196

😮 0.35958162

Fig. 13. POC prototype - single prediction user interface

4.2 Predicting the emotions of multiple phrases

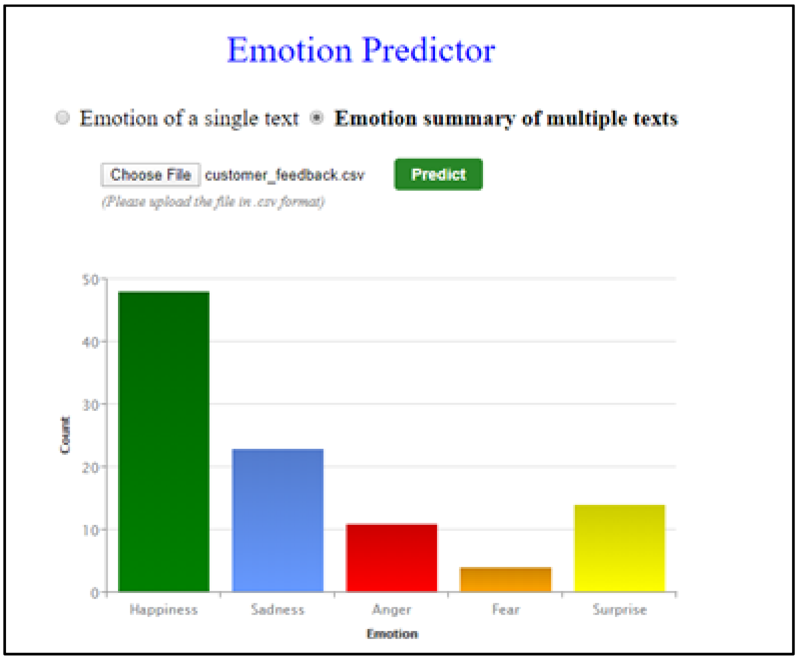
As shown by Fig. 14, a set of multiple texts can be submitted as comma-separated values. Then, the Java layer invokes the Python program. With the help of CNN, Python program predicts the emotion of each text and calculate the summary count.



Multiple texts uploaded in .csv format
Summary of emotion counts displayed as a chart
User Interface (JavaServer Page - JSP)
Multiple texts
Summary of emotion counts
Service Layer (Java Servlet)
Multiple texts
Summary of emotion counts
CNN (Python)

Fig. 14. POC prototype - multiple prediction flow

The final output is displayed to the user as shown by Fig. 15.



Emotion Predictor

○ Emotion of a single text ● **Emotion summary of multiple texts**

Choose File customer_feedback.csv

(Please upload the file in .csv format)

Fig. 15. POC prototype - multiple prediction user interface

This chapter explained how the proof of concept prototype was implemented to present the practical usage of the trained model. It explained how the trained model can be utilized to perform emotion detection of a single given phrase or a set of multiple phrases.

Chapter 5: Evaluation and Results

This chapter first explains the criteria followed to evaluate the trained CNN model along with the different types of evaluation metrics used. Then, the datasets used for the evaluation and the results are presented and discussed in detail.

5.1 Evaluation Metrics

In order to measure the performance of the trained deep learning model, classification metrics were calculated on a per-class basis. Before calculating the classification metrics, the confusion matrix was generated. A Confusion matrix [39] visualizes the predictions made by a model by comparing the actual and predicted classes. It can be considered as a base from which useful metrics can be generated. It shows how the model is confused when it makes predictions and gives insights into the errors that are being made by the model.

Table III displays the confusion matrix representation for binary class classification. Each row represents the instances of a predicted class, while each column represents the instances of an actual class.

TABLE III
Confusion Matrix representation for Binary Class Classification

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

The terms TP, FP, FN and TN have the following meanings.

- True Positive (TP): We predicted the class as ‘Positive’ and it is true
- True Negative (TN): We predicted the class as ‘Negative’ and it is true
- False Positive (FP): We predicted the class as ‘Positive’ and it is false. Also known as Type 1 Error

- False Negative (FN): We predicted the class as ‘Negative’ and it is false. Also known as Type 2 Error

The binary class representation of the confusion matrix can be extended to support multiclass classification as shown by Table IV. Here we are considering the predicted class as the positive class and all the other classes as negative.

TABLE IV
Confusion matrix representation for multi class classification

		Actual				
		Happiness	Sadness	Anger	Surprise	Fear
Predicted	Happiness	**TP**	FP	FP	FP	FP
	Sadness	FP	**TP**	FP	FP	FP
	Anger	FP	FP	**TP**	FP	FP
	Surprise	FP	FP	FP	**TP**	FP
	Fear	FP	FP	FP	FP	**TP**

Discussed below are the classification metrics which were derived from the confusion matrix.

5.1.1 Accuracy

Accuracy measures how often the classifier makes the correct prediction. It is the ratio between the number of correct predictions and the total number of predictions [40][41]. Model accuracy in predicting a class for a multiclass classifier can be calculated using two different methods.

- Micro: Provides a global metric by calculating the total number of times each class was correctly predicted and incorrectly predicted.
- Macro: Provides a per class based metric by calculating the values for each class independently, and finding their unweighted mean. Macro accuracy does not consider label imbalance.

Since the dataset used for evaluation was imbalanced, macro accuracy was used to calculate the overall model accuracy as shown by (1).

$$Accuracy\ in\ correctly\ predicting\ an\ emotion = \left(\frac{Total\ number\ of\ texts\ with\ emotion\ correctly\ predicted}{Total\ number\ of\ predictions}\right) \quad (1)$$

5.1.2 Precision

The precision of a class defines how trustable the result is when the model predicts that a point belongs to that class. It is the number of items correctly predicted as a certain emotion, out of the total items predicted as that emotion [40][41]. For each emotion class, precision was calculated separately. For example, precision in predicting happiness was calculated as shown by (2).

$$Precision\ in\ predicting\ happiness = \left(\frac{Number\ of\ texts\ with\ emotion\ correctly\ predicted\ as\ happiness}{Number\ of\ texts\ with\ emotion\ predicted\ as\ happiness}\right) \quad (2)$$

5.1.3 Recall

The recall of a class expresses how well the model is able to detect that class. It is the number of items correctly predicted as a certain emotion out of the total items with the emotion [40][41].

For each emotion class, recall was calculated separately. For example, recall in predicting happiness was calculated as shown by (3).

$$Recall\ in\ predicting\ happiness = \left(\frac{Number\ of\ texts\ with\ emotion\ correctly\ predicted\ as\ happiness}{Number\ of\ texts\ with\ emotion\ as\ happiness}\right) \quad (3)$$

Precision and recall are two useful metrics to measure the success of prediction when the classes are imbalanced. For a given class, the different combinations of precision and recall have the following meanings.

- High precision and high recall: The class is well handled by the model

- High precision and low recall: The class is not well detected by the model but is highly trustable when it does
- Low precision and high recall: The class is well detected but the model also includes points of other classes in it
- Low precision and low recall: The class is poorly handled by the model

5.1.4 F1-Score

F1 Score is the weighted average of precision and recall. It measures the effectiveness of identification when just as much importance is given to recall as to precision [40]. F1-Score is usually more useful than accuracy, especially when we have an uneven class distribution.

For each emotion class, f1-score was calculated separately. For example, f1-score in predicting happiness was calculated as shown by (4).

$$F1 - Score\ in\ predicting\ happiness = \left(\frac{2\ *\ Precision\ *\ Recall}{Precision\ +\ Recall}\right) \tag{4}$$

5.2 Evaluation Data

Of the complete dataset, 10% of the data was used for evaluation. The distribution on the dataset used for evaluation is shown by Fig. 16.

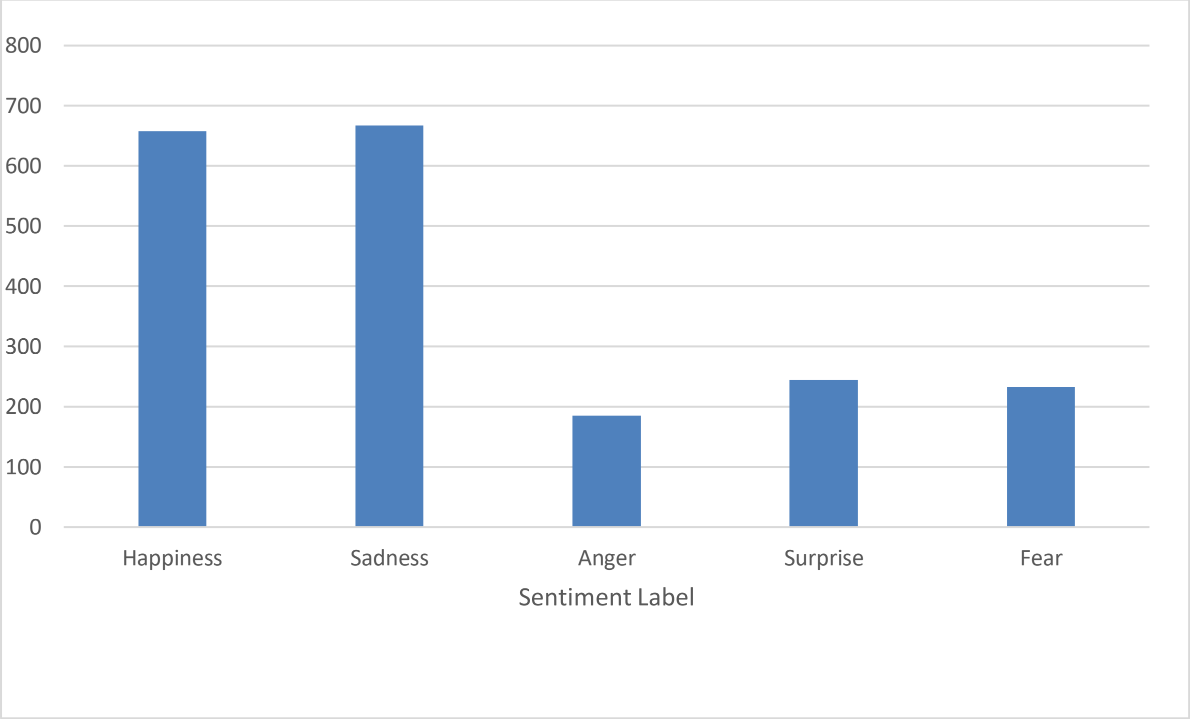


Fig. 16. Distribution of evaluation dataset

5.3 Evaluation Results

For each emotion class, actual and the prediction counts were calculated. Table V displays the confusion matrix generated based on the calculated values.

TABLE V
Confusion matrix

		Actual				
		Happiness	Sadness	Anger	Surprise	Fear
Predicted	Happiness	556	143	19	123	14
	Sadness	59	432	29	61	20
	Anger	5	18	124	3	5
	Surprise	30	45	6	55	7
	Fear	8	29	7	3	160

The overall model accuracy was calculated with the use of equation (1). The model accuracy in predicting a sentiment class is 67.67%.

Table VI displays the classification report of the classification metrics derived for each sentiment class with the use of confusion matrix by using the equations (2), (3) and (4).

TABLE VI
Classification report

Sentiment	Precision	Recall	F1-Score
Happiness	0.6503	0.8450	0.7350
Sadness	0.7188	0.6477	0.6814
Anger	0.8	0.6703	0.7294
Surprise	0.3846	0.2244	0.2834
Fear	0.7729	0.7769	0.7749

Fig. 27 provides a graphical representation of the evaluation metrics listed in the Confusion Matrix.

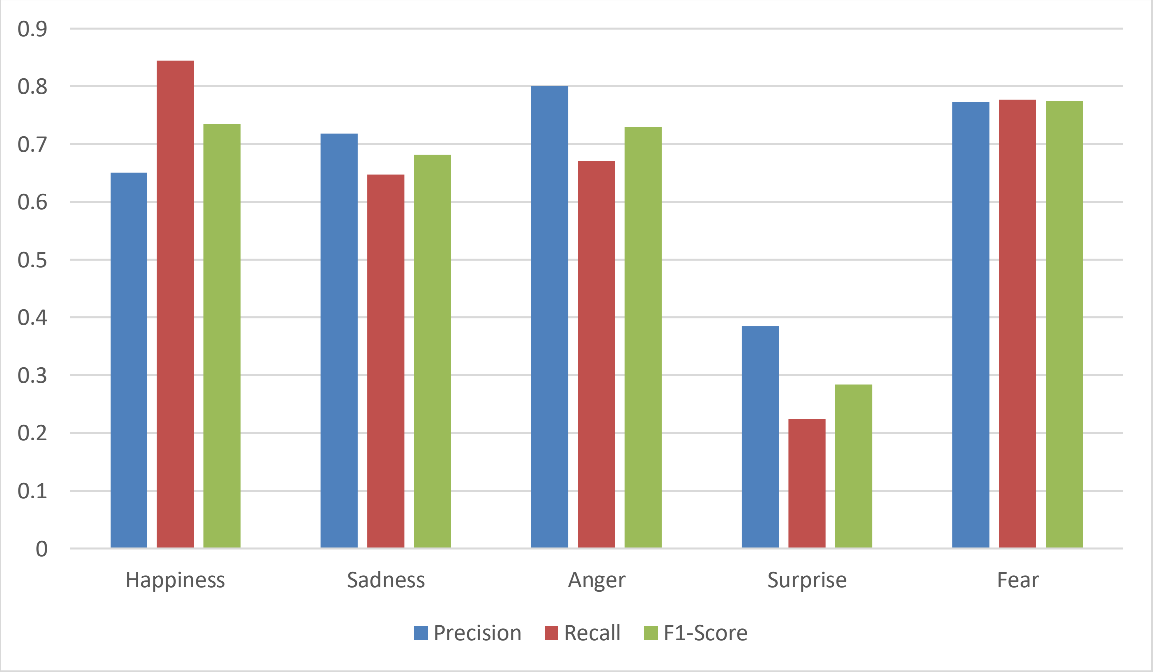


Fig. 17. Evaluation metrics distribution of sentiment classes

As per the classification report, sentiment class ‘Anger’ has the highest precision, which means that it has the highest trustability when the model predicts that a point belongs to that class. Sentiment class ‘Happiness’ has the highest recall. Which means out of all the classes the model well detects ‘Happiness’ sentiment class. “Fear” has the highest f1-score.

Apart from the sentiment class ‘Surprise’, all the other sentiment classes are having a precision and a recall over 0.5 which can be considered as a high precision and a recall. Therefore the four classes ‘Happiness’, ‘Sadness’, ‘Anger’ and ‘Fear’ are perfectly handled by the model and the class ‘Surprise’ is poorly handled by the model.

The three sentiments ‘Surprise’, ‘Fear’ and ‘Anger’ had a similar distribution of data when training the model. Their distribution was lower than the distributions used for the sentiment classes ‘Happiness’ and ‘Sadness’. However the accuracy in predicting ‘Fear’ and ‘Anger’ is high and similar to ‘Happiness’ and ‘Sadness’, but the accuracy in predicting ‘Surprise’ is low.

The emotion ‘Surprise’ is characterized by a physiological startle response following something unexpected. A surprise can be pleasant or unpleasant. Therefore this type of emotion can be positive, negative, or neutral making a classifier hard to differentiate this emotion with other emotions.

5.4 Summary

In the first section of this chapter, it provided the detailed information on the evaluation criteria followed to evaluate the trained CNN model along with the different types of evaluation metrics used. The evaluation metrics include macro accuracy, precision, recall and f1-score. Since the distribution of the dataset was imbalanced, these metrics were considered for the evaluation. Next, the results of the evaluation were presented and the observations were discussed in detail.

As per the analysis on the classification report, the model accuracy in predicting a sentiment class is 65-70%. The sentiment class ‘Anger’ has the highest trustability when the model predicts that a point belongs to that class and the model well detects the ‘Happiness’ sentiment class. The four classes ‘Happiness’, ‘Sadness’, ‘Anger’ and ‘Fear’, are perfectly handled by the model and the class ‘Surprise’ is poorly handled by the model.

Chapter 6: Conclusion

The study presented throughout the thesis aimed to identify whether we could utilize convolutional neural networks to perform multiclass sentiment analysis with higher accuracy and performance. Based on the literature review conducted it was identified that even though a significant amount of research has been done on binary class sentiment analysis using CNNs, there is a gap in performing multiclass sentiment analysis using the same. Therefore, in order to bridge the gap, the research under study uses a deep CNN for multi class sentiment classification for the sentiment classes 'Happiness', 'Sadness', 'Anger', 'Fear' and 'Surprise'. Based on the evaluation results, the proposed solution was able to predict an emotion of a given text with an accuracy of 65-70%.

The proposed approach accepts word embedding of a given sentence as the input, and executes multiple convolutions to generate feature maps. Max pooling is applied on the generated feature maps to prevent overfitting. Then the univariate vectors from each max pooling layer are concatenated together and a dropout layer is added to reduce overfitting. Finally, an activation function is applied to produce a probability distribution over the output sentiment classes.

The experimental results from the evaluation were cross validated with four main performance metrics accuracy, precision, recall and f1-score. The solution perfectly handles the four sentiment classes 'Happiness', 'Sadness', 'Anger' and 'Fear', and poorly handles the class 'Surprise'. As the emotion surprise can be pleasant or unpleasant, it can be positive, negative, or neutral making a classifier hard to differentiate this emotion with other emotions. The sentiment class 'Anger' has the highest trustability when the model predicts that a point belongs to that class. Out of all the classes, the model well detects 'Happiness' sentiment class.

Based on the evaluation of the results it can be concluded that the proposed approach addresses the research question whether we could utilize CNNs to perform multiclass sentiment analysis with a higher accuracy and performance. Out of the targeted five sentiment classes, the solution well predicts four classes and has an overall accuracy of 65-70%.

With the completion of the research two algorithms are introduced. The first algorithm trains and tests a CNN which supports multi class sentiment analysis. The second algorithm utilizes the trained CNN for predicting the emotion of a single text or set of multiple texts with a consolidated analysis. Both the algorithms are publicly available along with the sample outputs and the complete dataset used which will benefit researchers who are conducting studies in the said area.

The emotion ‘Disgust’ was not included in the research due to the lacking data. Therefore as for future work, studies can be conducted on CNNs accuracy and performance on detecting the emotion ‘Disgust’. Furthermore studies can be conducted to enhance the accuracy in classifying the emotion ‘Surprise’. The research was conducted with the Twitter data collected in social media domain. In future, the same research can be extended for sentiment analysis in other domains.

References

[1] J. Garcia-Garcia, V. Penichet and M. Lozano, "Emotion detection", Proceedings of the XVIII International Conference on Human Computer Interaction - Interacción '17, 2017. Available: 10.1145/3123818.3123852

[2] G. Balikas, S. Moura and M. Amini, "Multitask Learning for Fine-Grained Twitter Sentiment Analysis", *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '17,* 2017. Available: 10.1145/3077136.3080702

[3] A.García-Pablos, M. Cuadros and G. Rigau, "W2VLDA: Almost unsupervised system for Aspect Based Sentiment Analysis", *Expert Systems with Applications*, vol. 91, pp. 127-137, 2018. Available: 10.1016/j.eswa.2017.08.049

[4] V. A. and S. Sonawane, "Sentiment Analysis of Twitter Data: A Survey of Techniques", *International Journal of Computer Applications*, vol. 139, no. 11, pp. 5-15, 2016. Available: 10.5120/ijca2016908625

[5] A. D'Andrea, F. Ferri, P. Grifoni and T. Guzzo, "Approaches, Tools and Applications for Sentiment Analysis Implementation", *International Journal of Computer Applications*, vol. 125, no. 3, pp. 26-33, 2015. Available: 10.5120/ijca2015905866

[6] A. Chatterjee, U. Gupta, M. Chinnakotla, R. Srikanth, M. Galley and P. Agrawal, "Understanding Emotions in Text Using Deep Learning and Big Data", *Computers in Human Behavior*, vol. 93, pp. 309-317, 2019

[7] N. da Silva, L. Coletta, E. Hruschka and E. Hruschka Jr., "Using unsupervised information to improve semi-supervised tweet sentiment classification", *Information Sciences*, vol. 355-356, pp. 348-365, 2016

[8] K. Korovkinas, P. Danėnas and G. Garšva, "SVM and k-Means Hybrid Method for Textual Data Sentiment Analysis", Baltic Journal of Modern Computing, vol. 7, no. 1, 2019. Available: 10.22364/bjmc.2018.7.1.04

[9] Fathy, S., El-Haggar, N. and Haggag, M. (2017). A Hybrid Model for Emotion Detection from Text. International Journal of Information Retrieval Research, 7(1), pp.32-48

[10] F. Iqbal et al., "A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction", IEEE Access, vol. 7, pp. 14637-14652, 2019. Available: 10.1109/access.2019.2892852

[11] F. Khan, U. Qamar and S. Bashir, "A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet", *Knowledge and Information Systems*, vol. 51, no. 3, pp. 851-872, 2016

[12] S. Zhu, S. Li, Y. Chen and G. Zhou, "Corpus Fusion for Emotion Classification", Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 3287—3297, 20

[13] D. ZHOU, X. Zhang, Y. Zhou, Q. Zhao and X. Geng, "Emotion Distribution Learning from Texts", Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016. Available: 10.18653/v1/d16-1061

[14] M. Larsen, T. Boonstra, P. Batterham, B. O'Dea, C. Paris and H. Christensen, "We Feel: Mapping Emotion on Twitter", IEEE Journal of Biomedical and Health Informatics, vol. 19, no. 4, pp. 1246-1252, 2015. Available: 10.1109/jbhi.2015.2403839

[15]Q. Tul et al., "Sentiment Analysis Using Deep Learning Techniques: A Review", International Journal of Advanced Computer Science and Applications, vol. 8, no. 6, 2017. Available: 10.14569/ijacsa.2017.080657

[16] S. Rani and P. Kumar, "Deep Learning Based Sentiment Analysis Using Convolution Neural Network", *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3305-3314, 2018. Available: 10.1007/s13369-018-3500-z

[17] Z. Jianqiang, G. Xiaolin and Z. Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment Analysis", *IEEE Access*, vol. 6, pp. 23253-23260, 2018. Available: 10.1109/access.2017.2776930

[18] N. Nedjah, I. Santos and L. de Macedo Mourelle, "Sentiment analysis using convolutional neural network via word embeddings", *Evolutionary Intelligence*, 2019. Available: 10.1007/s12065-019-00227-4

[19] N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh and E. Cambria, "DialogueRNN: An Attentive RNN for Emotion Detection in Conversations", Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6818-6825, 2019. Available: 10.1609/aaai.v33i01.33016818

[20] S. Smetanin, Proceedings of the 13th International Workshop on Semantic Evaluation, 2019. Available: 10.18653/v1/s19-2034

[21] S. Lalitha, S. Tripathi and D. Gupta, "Enhanced speech emotion detection using deep neural networks", International Journal of Speech Technology, vol. 22, no. 3, pp. 497-510,
2018. Available: 10.1007/s10772-018-09572-8.

[22] D. Bertero and P. Fung, "A first look into a Convolutional Neural Network for speech emotion detection", *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. Available: 10.1109/icassp.2017.7953131

[23] S. Yoon, S. Byun, S. Dey and K. Jung, "Speech Emotion Recognition Using Multi-hop Attention Mechanism", ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019. Available: 10.1109/icassp.2019.8683483

[24] M. Lab, "DeepMoji", Deepmoji.mit.edu, 2019. [Online]. Available: https://deepmoji.mit.edu.

[25] Tone Analyzer", Tone-analyzer-demo.ng.bluemix.net, 2019. [Online]. Available: https://tone-analyzer-demo.ng.bluemix.net/

[26] "IBM Watson", Ibm.com, 2019. [Online]. Available: https://www.ibm.com/watson

[27] "Receptiviti - Understand the people who matter to your business.", Receptiviti, 2019. [Online]. Available: https://www.receptiviti.com/

[28] "Bitext. We help AI understand humans. Artificial Training for Chatbots", Bitext. We help AI understand humans., 2019. [Online]. Available: https://www.bitext.com/

[29] "Synesketch", Krcadinac.com, 2019. [Online]. Available: http://krcadinac.com/synesketch/

[30] "SuperDataScience", Superdatascience.com, 2019. [Online]. Available: https://www.superdatascience.com/blogs/the-ultimate-guide-to-self-organizing-maps-soms

[31] "Global Digital Report 2019 - We Are Social", *We Are Social*, 2019. [Online]. Available: https://wearesocial.com/global-digital-report-2019

[32] "Neural Network Embeddings Explained", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6

[33] "Individual Edition | Anaconda", *Anaconda*, 2020. [Online]. Available: https://www.anaconda.com/products/individual

[34] "Install TensorFlow with pip", *TensorFlow*, 2020. [Online]. Available: https://www.tensorflow.org/install/pip

[35] K. Team, "Keras: the Python deep learning API", *Keras.io*, 2020. [Online]. Available: https://keras.io/

[36] S. Team, "Spyder Website", *Spyder-ide.org*, 2020. [Online]. Available: https://www.spyder-ide.org/

[37] J. Brownlee, "Why One-Hot Encode Data in Machine Learning?", *Machine Learning Mastery*, 2020. [Online]. Available: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/

[38] "Natural Language Toolkit — NLTK 3.5 documentation", *Nltk.org*, 2020. [Online]. Available: https://www.nltk.org/

[39] "Understanding Confusion Matrix", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[40] "Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124

[41] "Multi-Class Metrics Made Simple, Part I: Precision and Recall", Medium, 2020. [Online]. Available: https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bdd

[42] M. Pontiki et al., "SemEval-2016 Task 5: Aspect Based Sentiment Analysis", *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016. Available: 10.18653/v1/s16-1002 [Accessed 20 January 2020].

[43] I. Im, B. Dunn, D. Lee, D. Galletta and S. Jeong, "Predicting the intent of sponsored search users: An exploratory user session-level analysis", *Decision Support Systems*, vol. 121, pp. 25-36, 2019. Available: 10.1016/j.dss.2019.04.001

[44] A. Adekotujo, J. Lee, A. Enikuomehin, M. Mazzara and S. Aribisala, "Bi-lingual Intent Classification of Twitter Posts: A Roadmap", *Advances in Intelligent Systems and Computing*, pp. 1-9, 2019. Available: 10.1007/978-3-030-14687-0_1

[45] R. Jagdale, V. Shirsat and S. Deshmukh, "Sentiment Analysis on Product Reviews Using Machine Learning Techniques", Cognitive Informatics and Soft Computing, pp. 639-647, 2018. Available: 10.1007/978-981-13-0617-4_61

[46] R. Amplayo and M. Song, "An adaptable fine-grained sentiment analysis for summarization of multiple short online reviews", Data & Knowledge Engineering, vol. 110, pp. 54-67, 2017. Available: 10.1016/j.datak.2017.03.009

Appendix I

GitHub Location

https://github.com/ruviniramawickrama/cnn-multiclass-sentiment-analysis/