OFFENSIVE MEME CLASSIFICATION THROUGH TEXT ANALYSIS USING EASY-OCR

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

BY

G. K. Mounica
(Roll No: 17331A0559)

M. Nivedita
(Roll No: 17331A0589)

K. Bharadwaj
(Roll No: 17331A0572)

K. Sandeep
(Roll No: 17331A0566)

Under the Supervision of

Dr. P. Srinivasa Rao
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MVGR COLLEGE OF ENGINEERING (Autonomous)

VIZIANAGARAM-535005, AP (INDIA)

(Accredited by NBA, NAAC, and Permanently Affiliated to Jawaharlal Nehru Technological University Kakinada)

JUNE, 2021

OFFENSIVE MEME CLASSIFICATION THROUGH TEXT ANALYSIS USING EASY-OCR

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

BY

G. K. Mounica
(Roll No: 17331A0559)

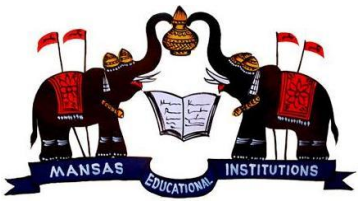
M. Nivedita
(Roll No: 17331A0589)

K. Bharadwaj
(Roll No: 17331A0572)

K. Sandeep
(Roll No: 17331A0566)

Under the Supervision of

Dr. P. Srinivasa Rao
Associate Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MVGR COLLEGE OF ENGINEERING (Autonomous)
VIZIANAGARAM-535005, AP (INDIA)
(Accredited by NBA, NAAC, and Permanently Affiliated to Jawaharlal Nehru Technological University Kakinada)

JUNE, 2021

CERTIFICATE

This is to certify that the project report entitled "**Offensive Meme Classification Through Text Analysis Using Easy-OCR**" being submitted by **G.K.Mounica, M.Nivedita, K.Bharadwaj, K.Sandeep** bearing registered numbers **13331A0559, 13331A0589, 13331A0572, 13331A0566** respectively, in partial fulfillment for the award of the degree of "**Bachelor of Technology" in Computer Science and Engineering** is a record of bonafide work done by them under my supervision during the academic year 2020-2021.

HOD CSE
DR. P. Ravi Kiran Varma
Associate Proffesor
Department of CSE
MVGR College of Engineering

Supervisor
Dr. P. Srinivasa Rao
Assistant Professor
Department of CSE
MVGR College of Engineering

External Examiner

ACKNOWLEDGEMENTS

We wish to express our sincerest and most profound gratitude to **Dr.P.Ravi KiranVarma,** Head of Department., Department of CSE, M.V.G.R. College of Engineering, Vizianagaram. He has always been a pillar of support for all our work both during the project and otherwise. It was a rare privilege working with him.

I thank **Dr. P. Srinivasa Rao**, Project In-charge and project guide, Assistant Professor, Department of CSE, M.V.G.R College of Engineering, Vizianagaram for helping us challenges our limits and not limit our challenges.

I also thank **Dr. K .V .L. Raju**, Principal for providing all provisions for successful completion of the project.

I thank the staff of Department of CSE , M.V.G.R. College of Engineering for helping us out in all the ways we needed and they could have.

G.K.Mounica(7331A0559)
M.Nivedita(17331A0589)
K.Bharadwaj(17331A0572)
K.Sandeep(17331A0566)

ABSTRACT

Traditional Media such as newspapers, televisions and radios are under constant supervision of the content they create and convey. Social media provides the internet users with less restrictions and monitoring over the facts and content they create. Although for most of the time these content mean no harm but some produce content due to the anonymity and freedom provided by social media. Most of these memes tend to be funny, but sometimes they might cross their limit to become offensive to specific individuals or groups, such memes could be referred to as troll memes. The main objective of our project is to classify the memes into offensive and non-offensive content through text analysis using Easy-OCR which internally uses LSTM model.

TABLE OF CONTENTS

List of Abbreviations

List of Figures

1. INTRODUCTION

1.1 INTRODUCTION TO PROJECT:

A meme is “an element of a culture or system of behavior passed from one individual to another by imitation or other non-genetic behaviors”. Memes come in a wide range of types and formats including, but not limited to images, videos, or twitter posts which has an increasing impact on social media communication. It is important to consider both modalities to understand the meaning or intention of the meme. Unfortunately, memes are responsible for spreading hatred in society, because of which there is a requirement to automatically identify memes with offensive content.

Social media networks provides an environment to use these networks as vulnerable hotspots to intrude /attack the victim’s privacy. So it is necessary to find a suitable action in order to detect and prevent offensive content in social media platforms. In this process of detection of offensive content, Machine learning can be helpful to detect meme texts that are used by bullies. Hence machine learning algorithms can generate a model to automatically detect offensive content from the memes. Cyberbullying emerged as a serious form of bullying due to recent growth of social media users.

1.2PROJECT OVERVIEW:

Social media are the interactive platforms that are in and around the daily life of most of the people .Memes has become integral part of daily life and they play a crucial role in sociopolitical, cultural and behaviour of the people .Memes are not only media for conveying information or disrupting the socio-political situation but, they serve as the main source of sharing humour and laughter. Internet humour can create a logical bonding between people and can act as a common platform for like minded people .Memes with colorful/even much dull background powered by splashy texts can create amazing sense of humour.

On the other hand, meme trolling is considered as online activism, creating awareness and new kind of marketing too. Troll is a person who starts the flame of insulting/ upsetting the feelings of another person on internet. This can be done by posting messages in the online community such as social media or even in private chats. Trolling has caused in one’s personal and professional life even. The content in any mainstream media are closely monitored but, for social media there are no such monitoring systems. Internet has opened the road to post anything and everything on social platforms. Offensive content memes are

spreading the emotion of fear, misguided phobia and they are misleading the population as they spend most of their time on the internet.

Considering the psychological, socio-political and social impacts of memes, we are proposing a deep learning based meme classifier that can differentiate the offensive one's from non-offensive counterparts.

1.3 REQUIREMENT SPECIFICATION:

HARDWARE SPECIFICATIONS:

Processor	:	i5
Ram	:	4GB
Hard Disk	:	500 GB

SOFTWARE SPECIFICATIONS:

Operating System	:	Windows 10
Code	:	Python

1.4 ENVIRONMENT SETUP :

1.4.1 Python Programming Language:

Python is an interpreted, high-level, general-purpose programming language. It's great as a first language because it is concise and easy to read. Use it for everything from web development to software development and scientific applications.

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index where, as of February 2020, it is the third most popular language (behind Java, and C). It was selected Programming Language of the Year in 2007, 2010, and 2018.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language

due to its comprehensive standard library. Python interpreters are available for many operating systems. Python is used extensively in the information security industry, including in exploit development.

Due to Python's user-friendly conventions and easy-to-understand language, it is commonly used as an intro language into computing sciences with students. This allows students to easily learn computing theories and concepts and then apply them to other programming languages.

1.4.2 Libraries used:

NumPy :

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

Using NumPy, a developer can perform the following operations − Mathematical and logical operations on arrays, Fourier transforms and routines for shape manipulation, Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

NumPy is often used along with packages like **SciPy** (Scientific Python) and **Mat−plotlib** (plotting library). This combination is widely used as a replacement for MatLab, a popular platform for technical computing.

Matplotlib :

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

EASYOCR:

EasyOCR is a python based OCR library which extracts the text from the image. Its a ready-to-use OCR with 40+ languages supported including Chinese, Japanese, Korean and Thai. It's an open source project licensed under Apache 2.0.

OPENCV:

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, text, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then whatever operations one can do in Numpy can be combined with OpenCV.

1.5 DATASET DESCRIPTION:

The dataset is considered on our own which is comprised of meme text data that are extracted from various meme image macros. These meme images are collected from various Social media platforms. Until now we have considered only 100 memes having text in it, for which few are used for training and few are used for testing the data.

2. LITERATURE SURVEY

2.1 LITERATURE SURVEY:

[1] Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, Paul Buitelaar, "Multimodel Meme Dataset (MULTIOFF) for Identifying Offensive Content in Image and Text", Language Resources and Evaluation Conference, May 2020.

Shardul *et al*[1], explained an approach on offensive content classification in memes based on images and text associated with it. MultiOFF dataset was used to train and evaluate a multimodal classification system for detecting offensive memes. Here the authors have used an automatically derived features through a pre-trained CNN, which is capable of classifying memes. LSTM and CNN are the models that have been used for this classification purposes. Logistic regression performs best in predicting the offensive meme category based on the text with high accuracy.. BiLSTM + VGG16 and CNNText + VGG16 performs best in predicting the offensive and not-offensive meme category for multimodel dataset. In this BiLSTM + VGG16 has achieved precision of 0.40, recall 0.44 and F1-score of 0.41. Whereas CNNText + VGG16 has achieved precision of 0.38, recall 0.67 and F-1 score of 0.48.

[2] Michele Tomaiuolo , Gianfranco Lombardo, Monica Mordonini , Stefano Cagnoni and Agostino Poggi, "A Survey on Troll Detection", February 2020.

Michele *et al*[2], explained that a troll is usually defined as somebody who provokes and offends people to make them angry, who wants to dominate any discussion or who tries to manipulate people′s opinions. This study discussed the problems of trolls in social media and presented various approaches for their detection. The first approach taken into consideration was to automatically analyse online contents through a natural language processing (NLP) approach. The second research direction involves the Social Network Analysis(SNA). SNA approach makes it possible to extract the information needed to assess the attitude of a user. Finally, another approach is through machine learning models. This model collects the features of trolls and legitimate users through the analysis of: writing style, sentiment, behaviours, social interactions, linked media and publication time. Support Vector Machine(SVM) classifier, that, once tested, has yielded a good identification percentage, approximately 90%.

[3] Tariq Habib Afridi, Aftab Alam, Muhammad Numan Khan, Jawad Khan, and Young-Koo Lee , "A Multimodel Memes Classification: A survey and Open-Research Issues", September 2020.

Tariq *et al*[3], explained that this article is performed on VL multimodel problems through which the memes classification could be done easily.Recent Machine Learning(ML) models are used to classify the memes. To determine the most effective classification model for any NLP task. Researchers have employed typical text classifiers such as SVM, kNN, Nave Bayes, ensemble classifiers such as Bagging. And for Multimodel-Linguistic Visual Classification RNN and LSTM models are used. In this LSTM has achieved highest test accuracy rate of 0.91 (91%) and training accuracy rate 0f 0.95 (95%).

[4] Yi Zhou, Zhenhao Chen, "Multimodal Learning for Hateful Memes Detection", November 2020.

Yi Zhou *et al*[4], explained this article have proposed a novel triplet- relation module for hateful memes prediction. It exploits the combination of image captions and memes to enhance Multimodel modeling. Triplet-Relation Network(TRN) is used which makes the relationship between image features and textual features and later help to detect the hateful meme. For this they used MSCOCO as pretraining data set for image captioner. It has 123000 images. They used 5000 images for testing and 5000 images for validation. They reported the performance of our models with the Area Under the Curve of the Receiver Operating Characteristic (AUROC). The AUROC of V+L model is 73.30 and V&L model is 71.88.

[5] Punyajoy Saha, Binny Mathew , Pawan Goyal and Animesh Mukherjee, "HateMonitors: Language Agnostic Abuse Detection in Social Media", January 2020.

Punyajoy *et al*[5], explained that this article detects the abusive text with pre-trained BERT and LASER sentence embeddings. Abusive language can be divided into hate speech, offensive language and profanity. Hate speech is a derogatory comment that hurts an entire group in terms of ethnicity, race or gender. Offensive language is similar to derogatory comment, but it is targeted towards an individual. SVM and Gradient Boosted trees are the models that considered first but Gradient boosted trees are often the choice for systems where features are pre-extracted from the raw data. In the category of gradient boosted trees, Light Gradient Boosting Machine (LGBM) is considered one of the most efficient in terms of memory footprint. Hence, they used LGBM as model for the downstream tasks in this competition with F1-score of 0.78.

[6] Manish Shetty M, Neelesh C.A, Pallavi Mishra," Offensive Text Detection using NLP", December 2020.

Manish *et al*[6], explained that this detects the offensive text in social media through Hybrid NLP approaches. They have used many hybrid approaches to improve the accuracy of the existing approaches. Naive Bayes , GRU with POS embeddings and character level CNN are the models that have been used for this approach in which CNN model have acheived the highest accuracy rate of 0.97 when compared to the other models.

[7] Sidharth Mehra, Mohammed Hasanuzzaman, "Detection of Offensive Language in Social Media Posts",May2020.

Sidharth *et al*[7], explained that this article had accelerated the detection of the posted offensive content so as to facilitate the relevant actions and moderation of these offensive posts. They had used the publicly available benchmark dataset OLID 2019 (Offensive Language Identification Dataset) for their research project. They contributed by making the training dataset balanced using the Random Under-sampling technique. They also performed a thorough comparative analysis of various Feature Extraction Mechanisms and the Model Building Algorithms. The final comparative analysis concluded that the best model came out to be Bidirectional Encoder Representation from Transformer (BERT). Thus their results outperformed the previous work achieving the Macro F1 score of 0.82 on this OLID dataset.

[8] Neo Cecillon, Vincent Labatut, Richard Dufour and Georges Linares, "Abusive Language Detection in Online Conversations by Combining Content- and Graph-Based Features",Frontier..inBigData,June2019.

Neo Cecillon *et al*[8], explained that they tackle the problem of automatic abuse detection in online communities. They took advantage of the methods that they previously developed to leverage message content and interactions between users, and create a new method using both types of information simultaneously. Support Vector Machine(SVM) model is used to distinguish abusive (*Abuse* class) and non-abusive (*Non-abuse* class) messages. The dataset contains nearly 4,029,343 messages in French, exchanged on the in-game chat of *SpaceOrigin*, a Massively Multiplayer Online Role-Playing Game (MMORPG). Among them, 779 have been flagged as being abusive by at least one user in the game, and confirmed as such by a human moderator. Thus they showed that the features extracted from our content- and graph-based approaches are complementary, and that combining them allows to sensibly improve the results up to 93.26 (*F*-measure).

3. THEORITICAL BACKGROUND

3.1 THEORITICAL BACKGROUND:

3.1.1Deep Learning :

Deep learning (also known as deep structured learning or differential programming) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi- supervised or unsupervised.

Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabelled data are more abundant than the labelled data. Examples of deep structures that can be trained in an unsupervised manner are neural history compressors and deep belief networks.

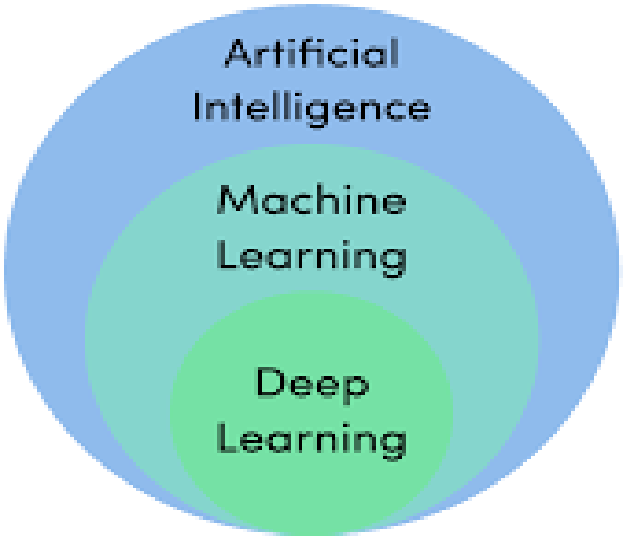


Figure 3-1 Deep Learning

Why Deep Learning ?

One of deep learning's main advantages over other machine learning algorithms is its capacity to execute feature engineering on it own. A deep learning algorithm will scan the data to search for features that correlate and combine them to enable faster learning without being explicitly told to do so.

Deep learning algorithms can be trained using different data formats, and still derive insights that are relevant to the purpose of its training. For example, a deep learning algorithm can uncover any existing relations between pictures, social media chatter, industry analysis, weather forecast and more to predict future stock prices of a given company.

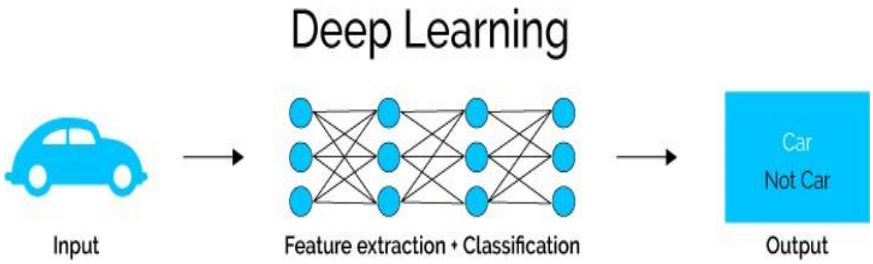


Figure 3-2 Deep Learning Model

Deep Learning techniques tend to solve the problem end to end, whereas Machine learning techniques need the problem statements to break down to different parts to be solved first and then their results to be combine at final stage. Deep learning architectures can be constructed with a greedy layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out which features improve performance

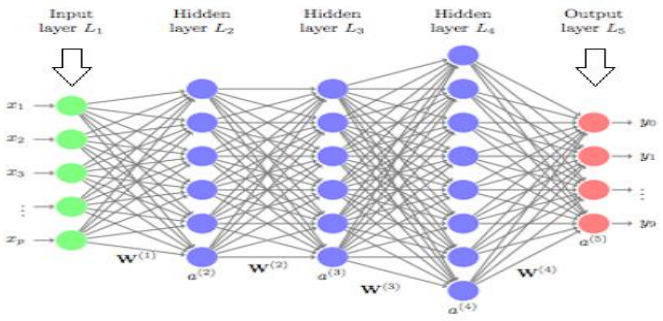


Figure 3-3 Hidden Layers

3.1.2 Difference Between Deep Learning & Machine Learning:

Machine learning is a subset, an application of Artificial Intelligence (AI) that offers the ability to the system to learn and improve from experience without being programmed to that level. Machine Learning uses data to train and find accurate results. Machine learning focuses on the development of a computer program that accesses the data and uses it to learn from themselves.

Deep Learning is a subset of Machine Learning where the artificial neural network, the recurrent neural network comes in relation. The algorithms are created exactly just like machine learning but it consists of many more levels of algorithms. All these networks of the algorithm are together called as the artificial neural network. In much simpler terms, it replicates just like the human brain as all the neural networks are connected in the brain, exactly is the concept of deep learning. It solves all the complex problems with the help of algorithms and its process.

Both machine learning and deep learning **mimic the way the human brain learns**. Its main difference is therefore the type of algorithms used in each case, although deep learning is more similar to human learning as it works with neurons. Machine learning usually uses **decision trees and deep learning neural networks**, which are more evolved. In addition, both can learn in a supervised or unsupervised way.

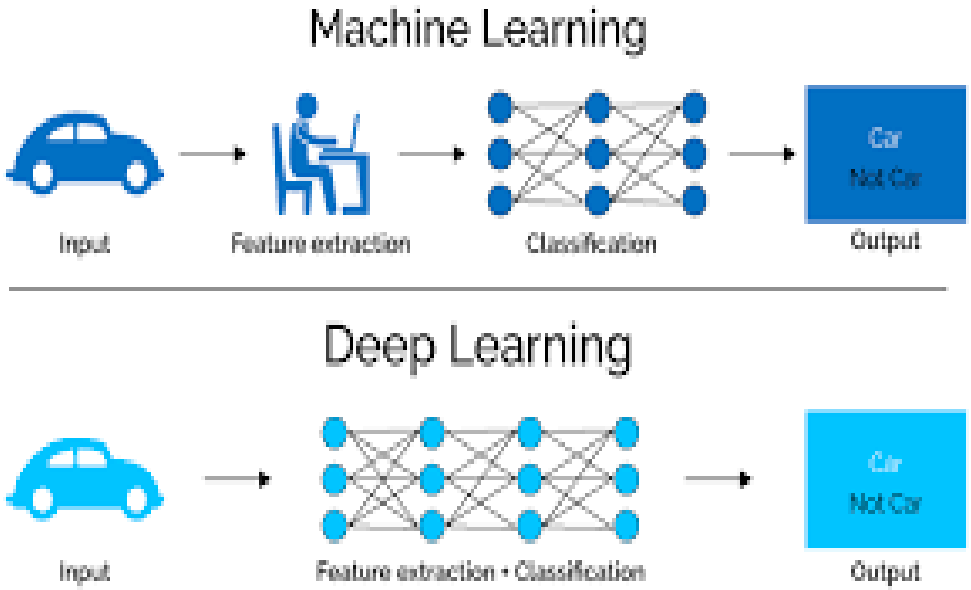


Figure 3-4 Machine Learning and Deep Learning Models

3.1.3 Recurrent Neural Networks:

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behaviour. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs.[1] This makes them applicable to tasks such as unsegmented, connected handwriting recognition[2] or speech recognition.[3][4]

The term “recurrent neural network” is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both classes of networks exhibit temporal dynamic behavior. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feedforward neural network, while an infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled.

Both finite impulse and infinite impulse recurrent networks can have additional stored states, and the storage can be under direct control by the neural network. The storage can also be replaced by another network or graph, if that incorporates time delays or has feedback loops. Such controlled states are referred to as gated state or gated memory, and are part of long short-term memory networks (LSTMs) and gated recurrent units. This is also called Feedback Neural Network.

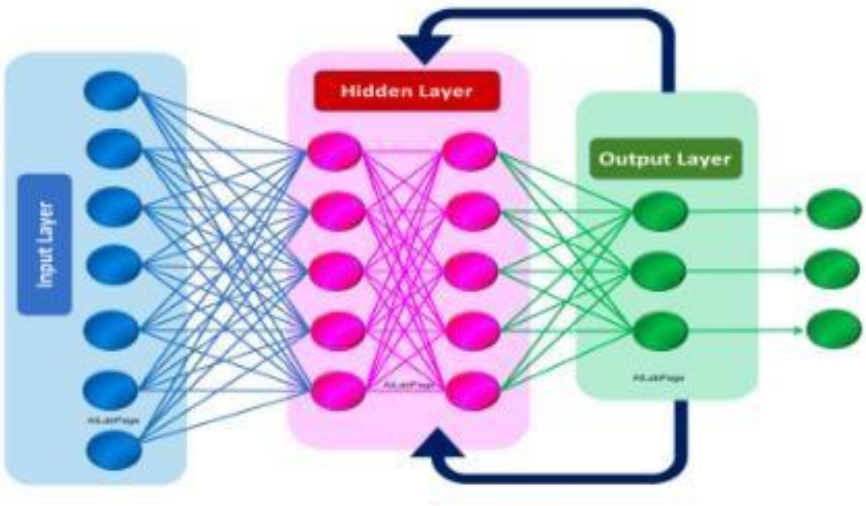


Figure 3-5 Recurrent Neural Network

Recurrent Neural Networks use backpropagation algorithm for training, but it is applied for every timestamp. It is commonly known as Back-propagation Through Time (BTT).

There are some issues with Back-propagation such as:

- Vanishing Gradient
- Exploding Gradient

Let us consider each of these to understand what is going on

Vanishing Gradient :

When making use of back-propagation the goal is to calculate the error which is actually found out by finding out the difference between the actual output and the model output and raising that to a power of 2.

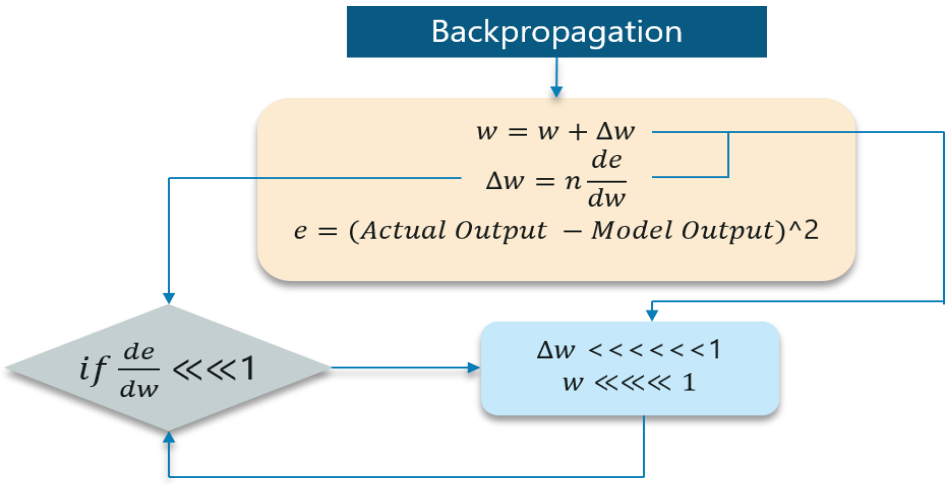


Figure 3-6 Vanishing Gradient(Error Calculation)

With the error calculated, the changes in the error with respect to the change in the weight is calculated. But with each learning rate, this has to be multiplied with the same.

So, the product of the learning rate with the change leads to the value which is the actual change in the weight.

This change in weight is added to the old set of weights for every training iteration as shown in the figure. The issue here is when the change in weight is multiplied, the value is very less.

Consider you are predicting a sentence say, “I am going to France” and you want to predict “I am going to France, the language spoken there is_”

A lot of iterations will cause the new weights to be extremely negligible and this leads to the weights not being updated.

Exploding Gradient :

The working of the exploding gradient is similar but the weights here change drastically instead of negligible change. Notice the small change in the diagram below.

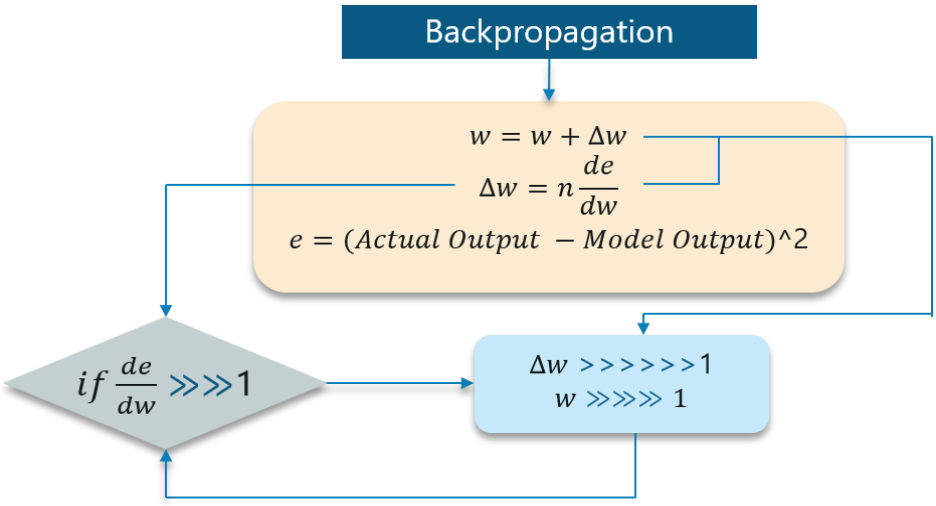


Figure 3-7 Exploding Gradient(Error Calculation)

3.1.4 Long Short-Term Memory Networks :

Long Short-Term Memory networks are usually just called “LSTMs”. They are a special kind of Recurrent Neural Networks which are capable of learning long-term dependencies.

What are long-term dependencies?

Many times only recent data is needed in a model to perform operations. But there might be a requirement from a data which was obtained in the past.

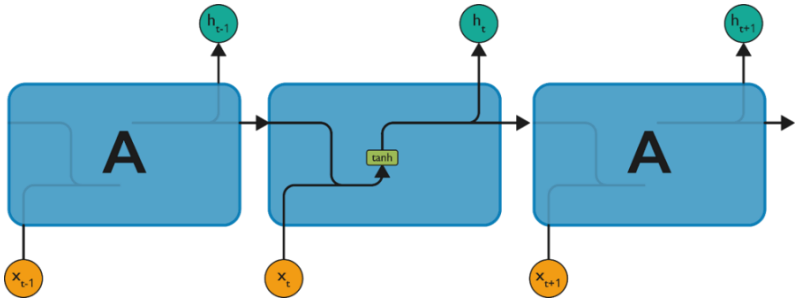
Let's look at the following example:

Consider a language model trying to predict the next word based on the previous ones. If we are trying to predict the last word in the sentence say "The clouds are in the sky".

The context here was pretty simple and the last word ends up being sky all the time. In such cases, the gap between the past information and the current requirement can be bridged really easily by using Recurrent Neural Networks.

So, problems like Vanishing and Exploding Gradients do not exist and this makes LSTM networks handle long-term dependencies easily.

LSTM have chain-like neural network layer. In a standard recurrent neural network, the repeating module consists of one single function as shown in the below figure:



As shown above, there is a tanh function present in the layer. This function is a squashing function**.** So, what is a squashing function?

It is a function which basically used in the range of -1 to +1 and to manipulate the values based on the inputs

Now, let us consider the structure of an LSTM network:

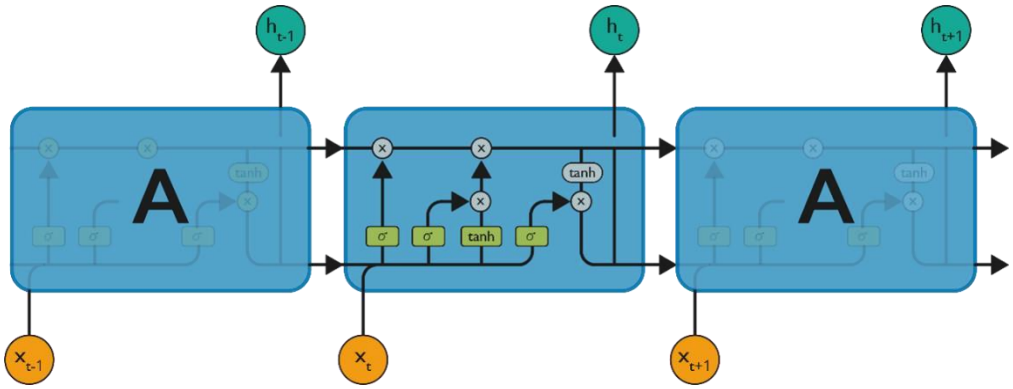


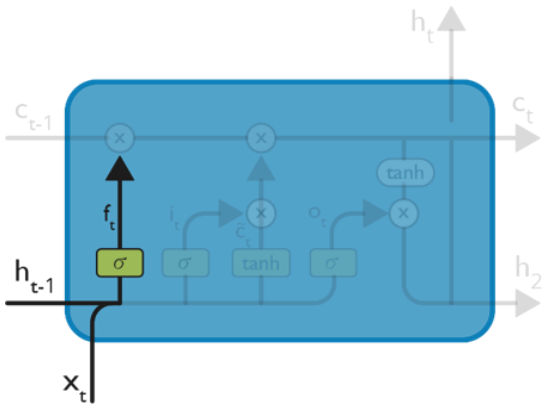
Figure 3-8 LSTM Network

As denoted from the image, each of the functions in the layers has their own structures when it comes to LSTM networks. The cell state is the horizontal line in the figure and it acts like a conveyer belt carrying certain data linearly across the data channel.

Let us consider a step-by-step approach to understand LSTM networks better.

Step 1:

The first step in the LSTM is to identify that information which is not required and will be thrown away from the cell state. This decision is made by a sigmoid layer called as forget gate layer.



The highlighted layer in the above is the sigmoid layer which is previously mentioned.

The calculation is done by considering the new input and the previous timestamp which eventually leads to the output of a number between 0 and 1 for each number in that cell state.

As typical binary, 1 represents to keep the cell state while **0** represents to trash it.

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

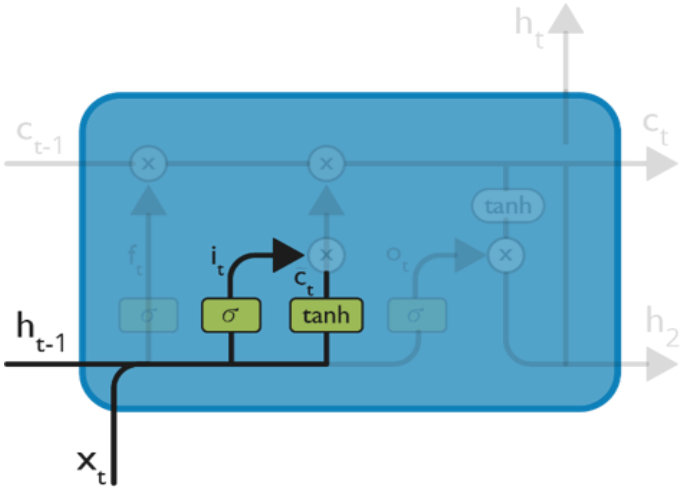
$w_f = Weight$
$h_{t-1} = Output\ from\ previous\ timestamp$
$x_t = New\ input$
$b_f = Bias$

Consider gender classification, it is really important to consider the latest and correct gender when the network is being used.

Step 2:

The next step is to decide, what new information we're going to store in the cell state. This whole process comprises of following steps:

- A sigmoid layer called the "input gate layer" decides which values will be updated.
- The tanh layer creates a vector of new candidate values, that could be added to the state.



$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$\tilde{c}_t = tanh(w_c[h_{t-1}, x_t] + b_c)$$

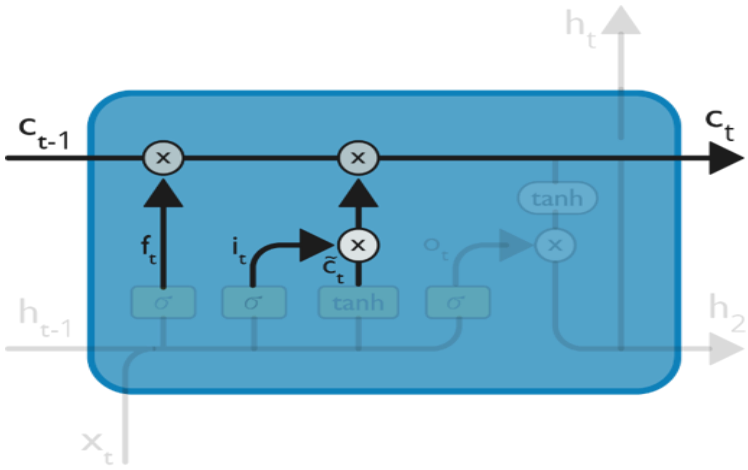
The input from the previous timestamp and the new input are passed through a sigmoid function which gives the value i(t). This value is then multiplied by c(t) and then added to the cell state.

In the next step, these two are combined to update the state.

Step 3:

Now, we will update the old cell state Ct−1, into the new cell state Ct.

First, we multiply the old state (Ct−1) by f(t), forgetting the things we decided to leave behind earlier.



$$c_t = f_t * c_{t-1} + i_t{}^* \tilde{c}_t$$

Then , we add **i_t* c˜_t.** This is the new candidate values, scaled by how much we decided to update each state value.

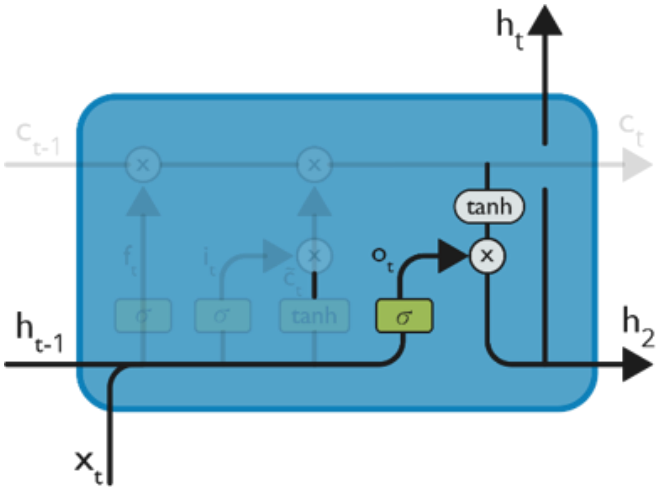
In the second step, we decided to do make use of the data which is only required at that stage. In the third step, we actually implement it.

In the language case example which was previously discussed**,** there is where the old gender would be dropped and the new gender would be considered.

Step 4:

We will run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh (push the values to be between −1 and 1)

Later, we multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.



$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(c_t)$$

The calculation in this step is pretty much straightforward which eventually leads to the output.

However, the output consists of only the outputs there were decided to be carry forwarded in the previous steps and not all the outputs at once.

Summing up all the 4 steps:

In the first step, we found out what was needed to be dropped.

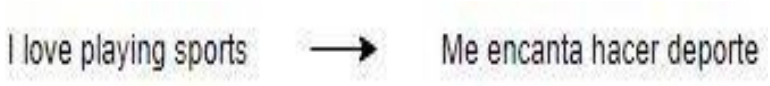
The second step consisted of what new inputs are added to the network.

The third step was to combine the previously obtained inputs to generate the new cell states. Lastly, we arrived at the output as per requirement.

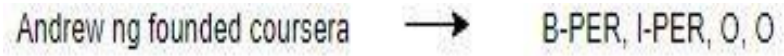
3.1.5 Sequence-to-Sequence (Seq2Seq) Modelling :

We can build a Seq2Seq model on any problem which involves sequential information. This includes Sentiment classification, Neural Machine Translation, and Named Entity Recognition – some very common applications of sequential information.

In the case of Neural Machine Translation, the input is a text in one language and the output is also a text in another language:



In the Named Entity Recognition, the input is a sequence of words and the output is a sequence of tags for every word in the input sequence:



There are two major components of a Seq2Seq model:

- Encoder
- Decoder

The Encoder-Decoder architecture is mainly used to solve the sequence-to-sequence (Seq2Seq) problems where the input and output sequences are of different lengths.

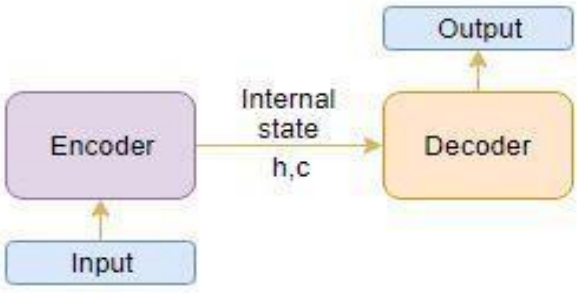


Figure 3-9 Encoder Decoder Architecture

The Encoder-Decoder architecture with recurrent neural networks has become an effective and standard approach for both neural machine translation (NMT) and sequence-to-sequence (seq2seq) prediction in general.

The key benefits of the approach are the ability to train a single end-to-end model directly on source and target sentences and the ability to handle variable length input and output sequences of text.

An Encoder-Decoder architecture was developed where an input sequence was read in entirety and encoded to a fixed-length internal representation.

A decoder network then used this internal representation to output words until the end of sequence token was reached. LSTM networks were used for both the encoder and decoder.

Generally, variants of Recurrent Neural Networks (RNNs), i.e. Gated Recurrent Neural Network (GRU) or Long Short Term Memory (LSTM), are preferred as the encoder and decoder components. This is because they are capable of capturing long term dependencies by overcoming the problem of vanishing gradient.

4. DESIGN AND IMPLEMENTATION

4.1 SYSTEM ARCHITECTURE:

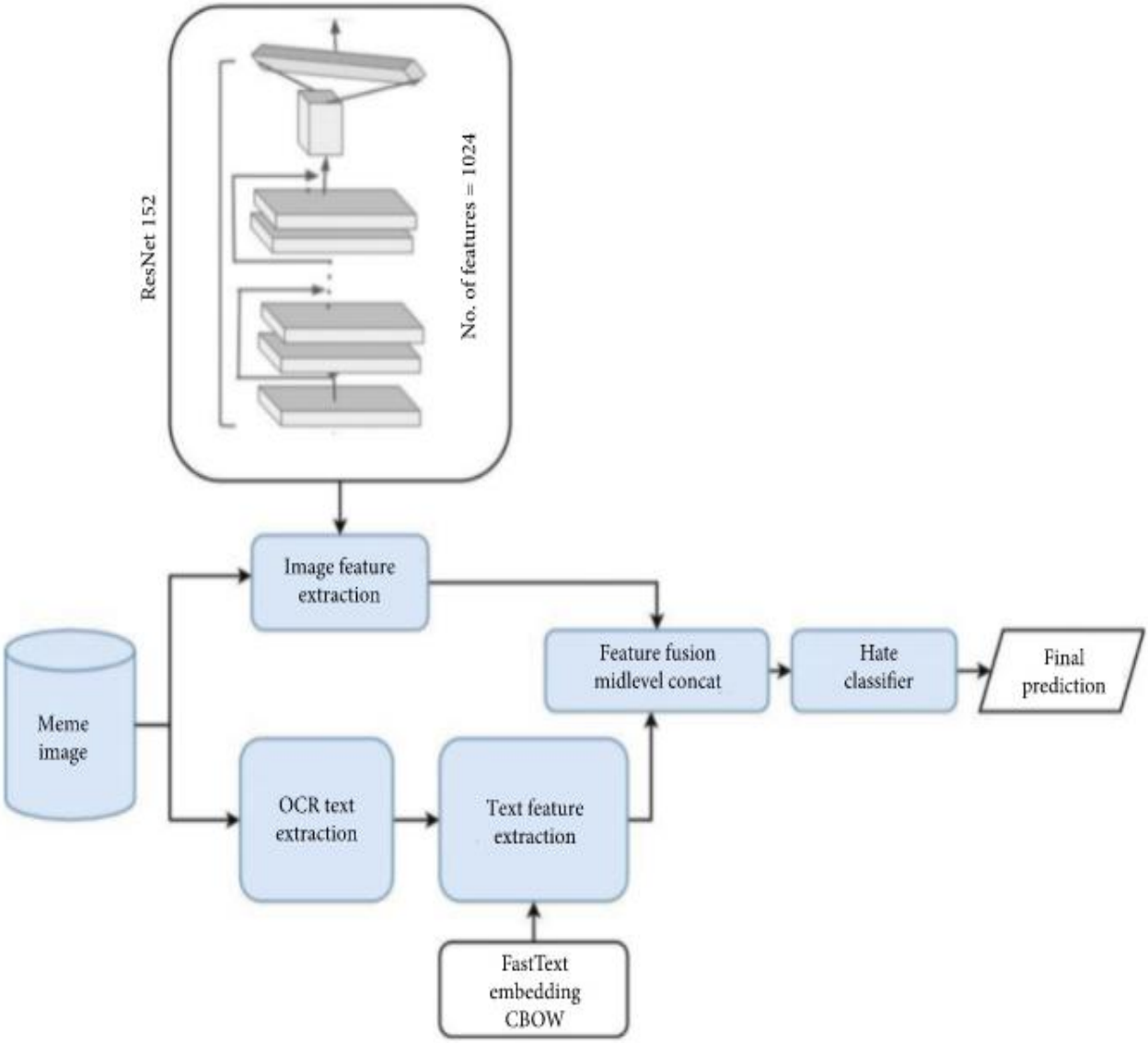


Figure 4-1 System Architecture

The proposed architecture will take a meme consisting of text. Now the text is extracted from the image using EASY-OCR and the extracted text is now validated to test the hatefulness and based on it, it will be either fall into offensive or non-offensive category. Easy-OCR internally uses LSTM model to detect the text containing hatefulness. Here the considered dataset consisting of set of memes that are splitted into training and testing categories which are used for validating and predicting the output.

4.2 PROPOSED SYSTEM:

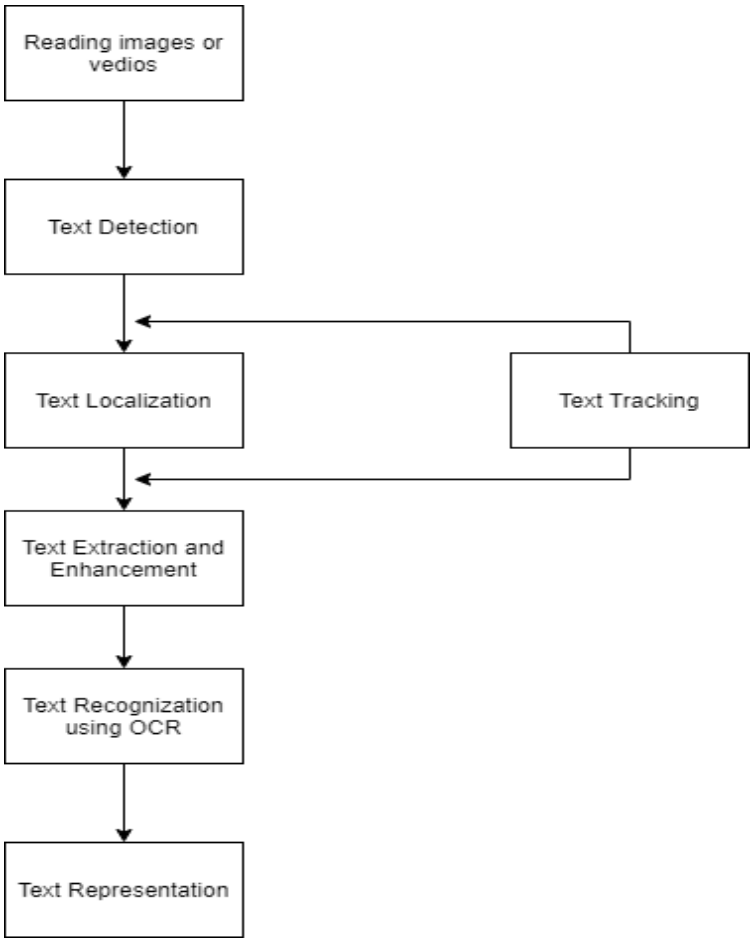


Figure 4-2 Flow Diagram

4.3 APPROACH USING EASY-OCR:

4.3.1 TESSERACT:

Tesseract is an open source text recognition (OCR) Engine, available under the Apache 2.0 license. It can be used directly or for programmers using an API to extract printed text from images. It supports a wide variety of languages. Tesseract doesn't have a built-in GUI, but there are several available from the 3rdParty pages. Tesseract is compatible with many programming languages and frameworks through wrappers that can be found here. It can be used with the existing layout analysis to recognize

text within a large document, or it can be used in conjunction with an external text detector to recognize text from an image of a single text line. Tesseract was an effort on code cleaning and adding a new LSTM model. The input image is processed in boxes (rectangle) line by line feeding into the LSTM model and giving output. In the image above we can visualize how it works.

4.3.2 OPENCV:

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, text, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then whatever operations one can do in Numpy can be combined with OpenCV.

4.3.3 EASYOCR:

EasyOCR is a python based OCR library which extracts the text from the image. Its a ready-to-use OCR with 40+ languages supported including Chinese, Japanese, Korean and Thai. It's an open source project licensed under Apache 2.0.

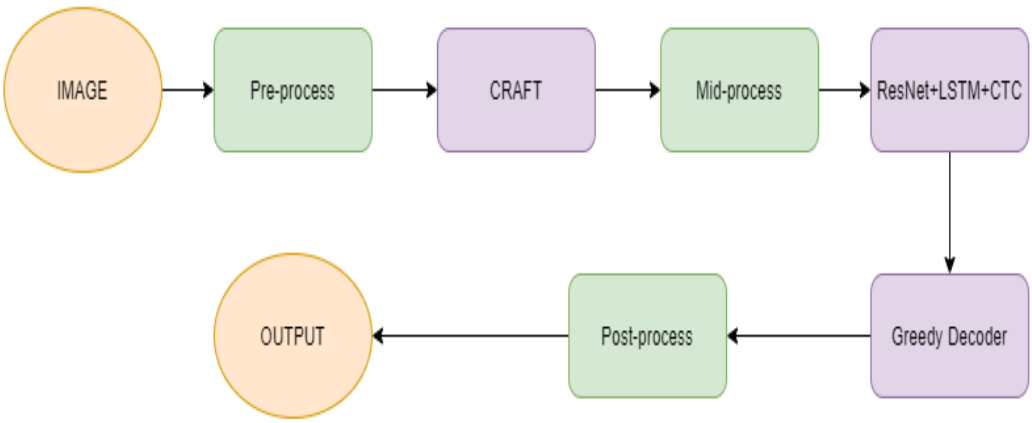


Figure 4-3 Easy-OCR

EasyOCR is built with Python and Pytorch deep learning library. The detection part is using the CRAFT algorithm and the recognition model uses CRNN. The sequencing labelling is performed by LSTM and CTC(Connectionist Temporal Classification). EasyOCR doesn't have much software dependencies, it can directly be used with its API.

- **CRAFT ALGORITHM:**

  CRAFT is a scene text detection method to effectively detect text area by exploring each character and affinity between the characters.

- **CRNN ALGORITHM:**

  CRNN is called Convolutional Recurrent Neural Network, is a convolutional recurrent neural network structure, used to solve image-based sequence recognition problems, especially scene text recognition problems. The biggest feature is that instead of cutting a single text first, the text recognition is transformed into a sequence-dependent learning problem based on image sequence.

- **LSTM ALGORITHM:**

  It is a special kind of RNN. Which is capable of learning long term dependencies thus treating the problem of short term dependencies of a simple RNN. It is not possible for a RNN to remember the to understand the context behind the input while we try to achieve this using a LSTM. LSTMs also have this chain like structure, but the repeating module has a different structure and has a similar control flow as a recurrent neural network. It processes data passing on information as it propagates forward.

- **CTC ALGORITHM:**

  CTC is called Connectionist Temporal Classification**.** CTC is meant for labelling the unsegmented sequence data with RNN.

4.4 IMPLEMENTATION REQUIREMENTS:

4.4.1 Python

- Need a Python version to use Jupiter and TensorFlow
- Link: https://repo.anaconda.com/archive
- We need to install system supported version - Anaconda3-2019.07-Windows-x86_64.exe
- While installing, we have to set path in environment variables or there is a check box to set path directly, we need to enable it.
- If we want to test it is installed correctly or not by opening command prompt and type python, then it shows the version of python - 3.7.3

4.4.2 Visual C++ Build Tools

- Tensorflow relies on C++ Build Tools and in next step we need to install CUDA, cuda needs Visual Studio to work successfully.
- Link: https://visualstudio.microsoft.com/vs/community
  After installing this, we need to install desktop development with C++, it is about 1.9GB

4.4.3CUDA and CUDNN for NVDIA

- CUDA and CUDNN are optional but if you have an nvidea GPU they speed up deep learning significantly
- Links:Cuda: 10.1- https://developer.nvidia.com/cuda-10.1-download-archive-base.
  Cudnn: 7.6.5- https://developer.nvidia.com/rdp/cudnn-archive
- It will check system compatibility and after that we need to select express version to install

4.4.4 Pytorch

- PyTorch is a prerequisite for the EasyOCR module. Its installation varies according to OS and GPU driver requirements.

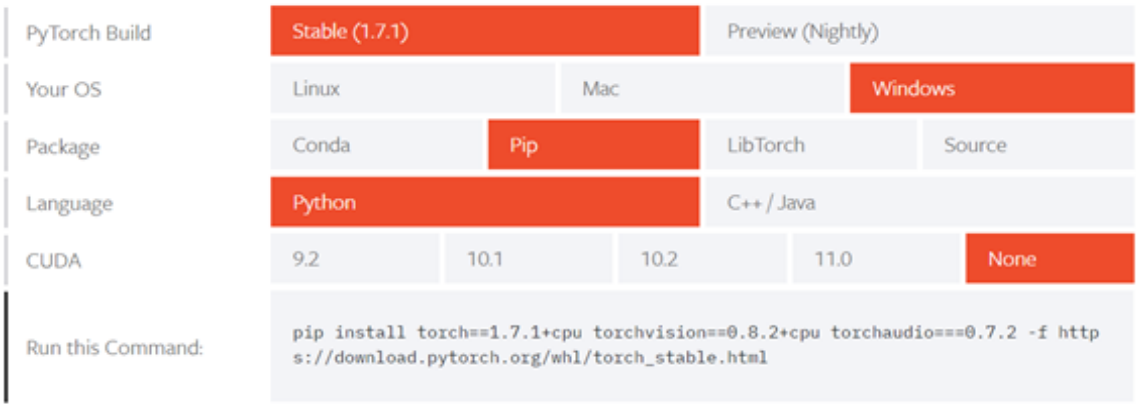


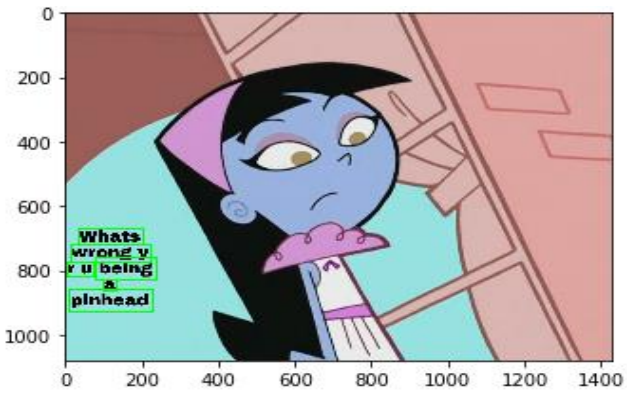
Figure 4-4 PyTorch Setup

4.4.5 EasyOCR

- pip install easyocr
- This command need to be run either in Jupiter notebook or in commandprompt for installing easyocr
- Now we need to import easyocr using
    - import easyocr

5. EXPERIMENTAL RESULTS

5.1 MODEL OUTPUT :



['Whats', 'wrong', 'y', 'r', 'u', 'being', 'a', 'pinhead']

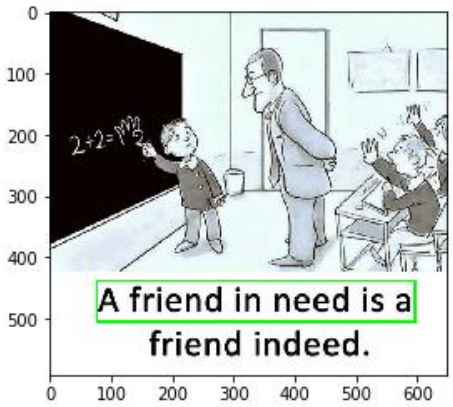
This MEME contains Abusive words. So it is an ABUSIVE MEME.

["Don't", 'be', 'a', 'pillock.']

This MEME contains Abusive words. So it is an ABUSIVE MEME.

['UNDERWATER', 'FIRE']

This MEME doesnot contains any Abusive words. So it is an NON-ABUSIVE MEME.

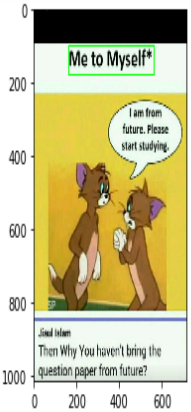



```
['A', 'friend', 'in', 'need', 'is', 'a', 'friend', 'indeed.', "2+2-'"]
```

This MEME doesnot contains any Abusive words. So it is an NON-ABUSIVE MEME.


```
['you', 'are', 'imbecile']
```

This MEME contains Abusive words. So it is an ABUSIVE MEME.




```
['Me', 'to', 'Myself*', 'Iam', 'from', 'future.', 'Please', 'start', 'studying:', 'SP', 'Jiaul', 'Islam', 'Then', 'Why', 'You',
'haventt', 'the', 'question', 'paper', 'from', 'future?', 'bring']
```

This MEME doesnot contains any Abusive words. So it is an NON-ABUSIVE MEME.

5.1 GRAPH :

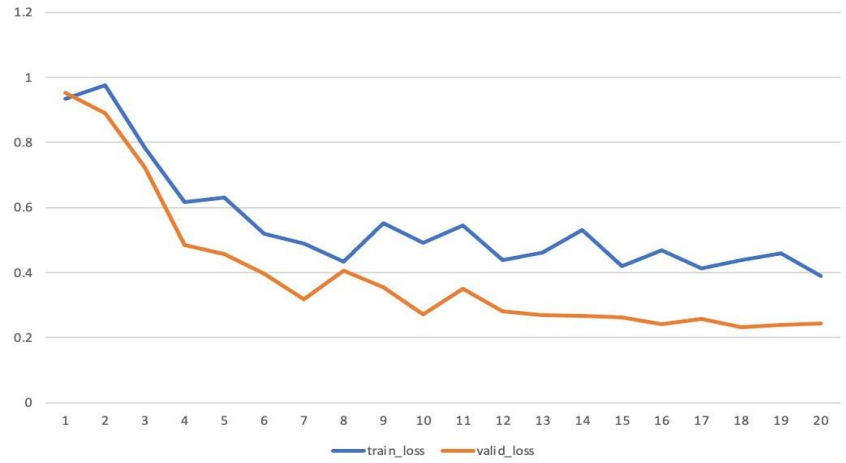


Figure 5-1 Loss Graph

Graph showing variation of loss during the course of training. Blue line shows decrease in training loss and orange line shows the variation of validation loss

6. CONCLUSION

5.1 CONCLUSION :

Memes on social media are one of the most popular ways to send false and hateful information to the masses. Memes poses as a medium which are powerful enough to disrupt politics, induce and spread humour. In the mean time they are capable of causing considerable harm to socio-political and personal health of individuals via trolls.Until now, we have read multiple papers regarding textual memes classification, natural language processing and RNN algorithms. There are multiple automatic text extractors with great capabilities and giving good results. Among them Easy-OCR is the one which we have used over here. We have successfully extracted the text from memes using this Easy-OCR and detected the offensive content based on the text extracted. Easy-OCR internally uses LSTM i.e deep learning algorithm for text prediction and extraction. We have successfully implemented this model to detect the offensive content from memes. There are few limitations of the model which can be improved in further work. First limitation is that it sometimes could not able to extract the words that are shorter in length(for eg: I), the other problem is it takes some time to generate a output if the meme text size is large enough, the other issue is it sometimes it detects as combination of two words that are shorter in length which is leading in a difficulty to detect the offensive content from it.

In future work, we will try to implement the same with better accuracy as after using RNN and LSTM the accuracy is bit low. There are many Offensive meme classifications available but all does not give appropriate result. Thus we will be using better algorithms to increase the effectiveness of the classification.

REFERENCES :

[1] Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, Paul Buitelaar,"Multimodel Meme Dataset for Identifying Offensive Content in Image and Text", Language Resources and Evaluation Conference, May 2020.

[2] Michele Tomaiuolo , Gianfranco Lombardo, Monica Mordonini , Stefano Cagnoni and Agostino Poggi , " A Survey on Troll Detection", Department of Engineering and Architecture, University of Parma, 43124 Parma, Italy, February 2020.

[3] Tariq Habib Afridi, Aftab Alam, Muhammad Numan Khan, Jawad Khan, and Young-Koo Lee, "A Multimodel Memes Classification: A survey and Open-Research Issues", Department of Computer Sceince and Engineering, Kyung-Hee-University, September 2020.

[4] Yi Zhou, Zhenhao Chen, "Multimodal Learning for Hateful Memes Detection", University of Maryland, Singapore, November 2020.

[5] Punyajoy Saha, Binny Mathew , Pawan Goyal and Animesh Mukherjee, "HateMonitors: Language Agnostic Abuse Detection in Social Media", Indian Institute of Technology, Kharagpur, West Bengal, India, January 2020.

[6] Manish Shetty M, Neelesh C.A, Pallavi Mishra," Offensive Text Detection using NLP", CSE Department, PES University, Bangalore, India, December 2020.

[7] Sidharth Mehra, Mohammed Hasanuzzaman, "Detection of Offensive Language in Social Media Posts", MSc Artificial Intelligence, Cork Institute of Technology,Munster Technological University, May 2020.

[8] Neo Cecillon, Vincent Labatut, Richard Dufour and Georges Linares, "Abusive Language Detection in Online Conversations by Combining Content- and Graph-Based Features", LIA, Avignon University, Avignon, France, June 2019.

[9] Prajakta Gaydhani, Virtee Parekh, Vaibhav Nagda, "Abusive Content Detection on Onine Social Media Forums", Department of Computer Science and Engineering, March 2019.

[10] Abdul Awal, Md. Shamimur Rahman, Jakaria Rabbi, " Detecting Abusive Comments in Discussion Threads using Naive Bayes", Department of Computer Sceince and Engineering, Khulna University of Engineering, October 2018.

[11] Cynthia Van Hee ,Gilles Jacobs ,Chris Emmery, Bart Desmet, Els Lefever, Ben Verhoeven, Guy De Pauw, Walter Daelemans, Véronique Hoste, "Automatic detection

of cyberbullying in social media text", https://doi.org/10.1371/journal.pone.0203794, October 2018.

[12] Haoti Zhong, Hao Li, Anna Squicciarini, Sarah Rajtmajer, Christopher Griffin, David Miller, Cornelia Caragea, "Content-Driven Detection of Cyberbullying on the Instagram Social Network", Dept. of Electrical Engineering, Dept of Mathematics, Dept of Computer Science, Pennsylvania State University, United States Naval Academy, University of North Texas, November 2016.

[13] Harish Yenala, Ashish Jhanwar, Manoj K. Chinnakotla & Jay Goyal, "Deep learning for detecting inappropriate content in text", International Journal of DataScience and Analytics, December 2017.

[14] Rogers Pelle, Cleber Alcantara, Viviane P. Moreira, " A Classifier Ensemble for Offensive Text Detection", October 2018.

[15] Busra Mutlu, Merve Mutlu , Kasım Oztoprak, Erdogan Dogdu, " Identifying Trolls and Determining Terror Awareness Level in Social Networks using Scalable Frameworks", Department of Computer Engineering, KTO Karatay University, December 2016.

[16] Youssef Kishk, " NLP-Social-Media-Offensive-language-detection", Department of Computer Science and Engineering, January 2019.

[17] Hao Chen, Susan McKeever, Sarah Jane Delany, " Abusive Text Detection Using Neural Networks ", School of Computer Science, Dublin Institute of Technology, Ireland, November 2017.

[18] Cheng, J. Danescu-Niculescu-Mizil, C. Leskovec, " Antisocial behavior in online discussion communities.", In proceedings of the Ninth International AAAI Conference on Web and Social Media, Oxford, UK, 26–29 May 2015.

[19] Atanasov, A., Morales, G.D.F.; Nakov, "Predicting the Role of Political Trolls in Social Media", 2019.

[20] Machová, K.; Kolesár, "Recognition of Antisocial Behavior in Online Discussions", In International Conference on Information Systems Architecture and Technology; Springer: Cham, Switzerland, 2019.