Multimodal Assessment of Abusive Content in Tamil Language: Deep Learning Approach

A PROJECT REPORT

Submitted by

Kaushik M - (CB.EN.U4AIE19036)
Prasanth S N - (CB.EN.U4AIE19046)
R Aswin Raj - (CB.EN.U4AIE19050)
Vijai Simmon S - (CB.EN.U4AIE19068)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE)

Center for Computational Engineering and Networking

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641112 (INDIA)

MAY - 2023

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

AMRITA
VISHWA VIDYAPEETHAM
श्रद्धावान् लभते ज्ञानम्
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF THE UGC ACT, 1956

BONAFIDE CERTIFICATE

This is to certify that the thesis entitled **Multimodal Assessment of Abusive Content in Tamil Language: Deep Learning Approach** submitted by **Kaushik M (CB.EN.U4AIE19036)**, **Prasanth S N (CB.EN.U4AIE19046)**, **R Aswin Raj (CB.EN.U4AIE19050)**, and **Vijai Simmon S (CB.EN.U4AIE19068)** for the award of the **Degree of Bachelor of Technology** in the **Computer Science and Engineering (Artificial Intelligence)** is a bonafide record of the work carried out by them/him/her under my guidance and supervision at Amrita School of Computing, Coimbatore.

Dr. Premjith B **Dr. Sowmya V** **Dr. Jyothish Lal G**

Project Guides

Dr. K.P.Soman
Professor and Head
CEN

Submitted for the university examination held on ——

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Contents

Acknowledgement

We like to express our sincere gratitude towards Dr Premjith, Dr Sowmya V and Dr Jyothish Lal G for their constant and undivided support and guidance throughout this project. We would also like to extend our gratitude towards Dr Soman K P, Associate Dean and the Head of Department for Center of Computational Engineering and Networking department who gave this wonderful opportunity to work on this project. We also like to acknowledge Amrita Vishwa Vidyapeetham management for providing us the required resources to complete this project on time.

List of Figures

List of Abbreviations

CFS	Correlation based Feature Selection
SMOTE	Synthetic Minority Oversampling Technique
OCSVM	One Class Support Vector Machine
IF	Isolation Forest
ML	Machine Learning
DL	Deep Learning
SVM	Support Vector Machine
RF	Random Forest
LDA	Linear Discriminant Analysis
NNet	Neural Network
EER	Equal Error Rate
TF-IDF	Term Frequency Inverse Document Frequency
MFCC	Mel Frequency Cepstral Coefficients
STFT	Short Time Fourier Transform
GRU	Gated Recurrent Units
LSTM	Long-Short Term Memory

Abstract

The social media platforms are not offering effective regulations for abusive contents. Even today, it is done by manual monitoring of abusive behaviour on social media which is impractical and not suitable for today's world. Till date, only user reporting mechanisms have been implemented in platforms like Twitter. These measures are not scalable and not a long-term solution to this problem. So, robust approaches are needed for abusive language detection in a multi domain and multilingual environment, which will also enable the implementation of effective tools that could be employed to support both monitoring and content moderation activities such as automatic moderation and flagging of potentially hateful users and posts, also for guaranteeing a better compliance to governments demands to counteract the phenomenon. In this work we have created our own data-set for Tamil language for abusive content detection. We have experimented with Multi-modal approaches and this work showcases our experiments and results we obtained on the same.

Chapter 1

Introduction

Abusive language is becoming a pertinent problem because, People use different modes of content in social media platforms such as Instagram, Facebook, Twitter and YouTube. Many people all around the world share their reviews, opinions and videos through online platforms in different modalities such as text, audio and video. These types of modalities of data are too difficult to process and analyse because of ambiguity and anonymity. Moreover, such types of data are flexible for users. It is easy to connect globally with less effort. This is taken as an advantage and exploits the harmful content such as hate or offensive speech or in other forms of abusive language. For example, around 3.7 million videos are uploaded to YouTube every day. That's around 271,330 hours of video content based on the average length of 4.4 minutes, it's impractical to monitor the abusive content in the modalities. Till date, only user report mechanisms have been implemented in platforms like Twitter. These measures are not a scalable and long-term solution to this problem. The Artificial Intelligence includes Deep Learning or Machine Learning model should extract the features from all types of modalities for better analysis and result.

In this online era, there is an ever-increasing number of malevolent actors who use social media to harm others. Therefore, it is important to stop online hate speech from becoming viral and contributing significantly to serious crimes against minorities or vulnerable groups in a variety of languages and circumstances .

In this online era, there is an ever-increasing number of malevolent actors who use social media to harm others. Therefore, it is important to stop online hate speech from becoming viral and contributing significantly to serious crimes against minorities or vulnerable groups in a variety of languages and circumstances[4]. The goal of our task here is to find the abusive speech in the dataset created for Tamil language. We created our own dataset using the approach mentioned in the paper[2]. Our dataset contains videos taken from You Tube and manually labelled as Abusive or Non-Abusive. The transcript of the video is also manually created in Tamil. Our dataset contains 47-Abusive videos and 41-Non-Abusive Videos. We pursued this project because there is no work on multi-modal Tamil abusive language detection dataset and no AI models have been for the same.

This work focuses on developing a multi-modal architecture which can use all the three modalities to classify the given input video either as Abusive or Non-Abusive by initially making a comparative study on models trained for each of the modalities. The methodology and results are explained and depicted clearly in the upcoming sections. The rest of the paper is organized as follows. Section II describes about the dataset that we have collected and annotated; Section 3 briefly describes about the methodology that we carried out. In this section we have briefly explained about the models used and the feature extraction techniques that we used for each modality; Section IV shows the best results that we obtained among all the methods described in Section 3; Section V summarize the findings of our work and Section VI concludes our paper with future works. This section explains about our future ideas and the process we like to carry on for better development and deployment of the model.

1.1 Literature Survey

In today's world, the use of social media is growing exponentially along with content being shared in these platforms. Many people all around the world share their content in the form text, audio and video. These platforms also became a tool for malevolent actors to spread abusive contents pointing towards a particular person or a group in various means of content. Most of the existing approaches either use text or audio or video modality. There are significant works done in the domain of the multimodal sentiment analysis but there are significantly less number of works done in the domain of abuse detection. Few works involve multimodal approaches which incorporates the text, audio and video classification into a single classification model.

In paper[1], the authors provides a brief summary about various popular approaches that have been used to extract features from multimodal videos in addition to a comparative analysis between the most popular multimodal sentiment analysis benchmark datasets. According to this paper, MARNN model performed better on MOSI dataset for the binary classification which used features extracted from the text, audio and video using feature extraction techniques like word2vec, openSMILE and 3D-CNN respectively. In paper[4], the authors compared various deep learning-based models on multimodal video sentiment analysis. In paper[5], the author used audio modality along with the text modality to incorporate the emotional attributes of the content and which in return boosts the performance of the abusive content classification models. In paper[3], the authors listed several challenges and open problems on this area.

All the existing multimodal abuse detection are done on selective languages like English. Our work focuses on detecting the abusive content in Tamil language. We have collected and annotated the data as per the steps mentioned in the paper[2].In paper[**6**] researchers used CREENDER tool to create a multi model dataset of images and abusive comments. This paper details the tool CREENDER as well as to raise the awareness on the images and comments spotted online, also present some statistics on the dataset. During annotation, the most common sematic categories were assigned. Finally, the research is on, if there is correlation between the subject of a picture and the offensive message associated with it. Whenever, it is necessary to create datasets that on the hand compliant with privacy issues and on the other hand are-user friendly. This annotation tool, which can be can be employed with user-specific images taken from any freely available dataset is meant to overcome the privacy issue. 95 Italian high school students who were participating in activities with schools on the subject of cyberbullying helped build the tool. Therefore, CREENDER was employed to spread awareness about the photographs shared online and the objectionable material in some comments, facilitating the development of a dataset to investigate these occurrences.

In this approach, participants assess if photos could potentially elicit an objectionable statement, leave a potential comment, and also assign to a trigger group. Subject types for pictures annotated with ‘Yes’ (i.e. triggering a comment) and ‘No’.

Then comparing the two distributions, that are reported, by applying the 2 test (N = 4, 218), showing a statistically significant difference between the two distributions of categorial variables (p greater than .001). Further details about the dataset collection and challenges faced are covered in the upcoming chapters.

1.2 Problem statement

The problem statement for the project ”Multi-modal Assessment of Abusive Content in Tamil Language: Deep Learning Approach” is to develop a system that can accurately detect and classify abusive content in the Tamil language across various modalities, including text, images, and audio. The system should be able to identify various types of abusive content such as hate speech, cyberbullying, and harassment, and classify them into different categories. The project aims to use deep learning techniques to train a multi-modal model that can handle the complexity and nuances of the Tamil language, including the use of slang and local dialects. The ultimate goal of this project is to develop a tool that can assist social media platforms and other organizations in identifying and removing abusive content in the Tamil language, thereby creating a safer and more inclusive online environment for Tamil community.

1.3 Objectives

The main objectives of the work is as follows:

- To develop a dataset for multi-modal Tamil abusive language detection.

- To develop deep learning based models for detecting videos containing abusive contents by incorporating video, speech and text modalities.

- To compare the performance of different models by considering different combinations of features extracted from three modalities.

Chapter 2

Background

2.1 Machine Learning Models

2.1.1 Logistic Regression

One of the most often used Machine Learning algorithms, within the category of Supervised Learning, is logistic regression. Logistic regression is commonly used for classification problems and prediction problems. There are three types of logistic regression, out of which we used binary logistic regression for our classification problem to find Abusive or Non-Abusive in text modality.

2.1.2 SVM - Support Vector Machine

One of the most well-liked supervised learning algorithms, Support Vector Machine, or SVM, is used to solve Classification and Regression issues. However, it is largely employed in Machine Learning Classification issues. The SVM algorithm's objective is to establish the optimal line or decision boundary that can divide n-dimensional space into classes Abusive and non-Abusive, allowing us to quickly classify fresh data points in the future. A hyperplane is the name given to this optimal decision boundary.

2.1.3 SVC - Support Vector Classifier

The supervised machine learning technique known as SVC, or Support Vector Classifier, is frequently used for classification problems. SVC separates the data into two classes Abusive or non-Abusive by mapping the data points to a high-dimensional space and then locating the best hyperplane.

2.1.3.1 SVC-RBF Kernel

Radial Basis Kernel (RBF)is a kernel function that is used in machine learning to find a non-linear classifier used in SVC. The RBF kernel, which is the default kernel in the sklearn SVM classification method, contains the following formula: where gamma can be manually changed and must be greater than 0.

2.1.4 Random Forest

An ensemble learning technique for classification and other tasks called random forests or random decision forests works by building a large number of decision trees during the training phase. The class that the majority of the trees choose is the output of the random forest for classification problems. The mean or average forecast of each individual tree is returned for regression tasks. The tendency of decision trees to over-fit their training set is corrected by random decision forests. Although they frequently outperform decision trees, gradient boosted trees are more accurate than random forests. However, their effectiveness may be impacted by data characteristics. As our project states with classification problems, the output of the random forest is the class of Abusive or Non-Abusive selected by most trees.

2.1.5 Stochastic Gradient

A method for optimising an objective function with sufficient smoothness qualities (such as differentiable or sub-differentiable) is stochastic gradient descent, or SGD for short. In order to drastically minimise calculations, SGD chooses one data point at random from the whole data set for each iteration. Another typical practice is known as ”mini-batch” gradient descent, which involves sampling a small number of data points rather than just one at each step.

2.1.6 XGBoost

Extreme Gradient Boosting (XGBoost) is a distributed, scalable gradient-boosted decision tree (GBDT) machine learning framework. The top machine learning library for regression, classification, and ranking issues, it offers parallel tree boosting. Regularisation is not incorporated in gradient boosting, but it is in XGBoost, a regularised version of gradient boosting where L2 or L1 regularisation is already in place. As it is a classification problem in our case it is used to find Abusive or Non-Abusive.

2.1.7 K Neighbours Classifier

The k-nearest neighbours algorithm, sometimes referred to as KNN or k-NN, is a supervised learning classifier that employs proximity to produce classifications or predictions about the grouping of a single data point. Radius Neighbors Classifier provides classification based on all neighbourhood points within a given radius, r, of target point, t, whereas K Neighbors Classifier implements classification based on voting by nearest k-neighbors of target point, t. We used the K Neighbors classifier to classify Abusive or non-Abusive in Audio Modality.

2.1.8 Decision Tree Classifier

The non-parametric supervised learning approach used for classification and regression applications is the decision tree. It is a supervised machine learning algorithm that employs guidelines to make choices, much like how people do. The idea behind Decision Trees is to repeatedly partition the dataset until all the data points that belong to each class are isolated by using the dataset characteristics to produce yes/no questions. It is organised hierarchically and has a root node, branches, internal nodes, and leaf nodes.

2.2 Deep Learning Models

2.2.1 CNN - Convolutional Neural Networks

CNN is a feed-forward neural network that is generally used to analyse visual images by processing data with grid-like topology. It's also knows as a ConvNet. The convolution operation

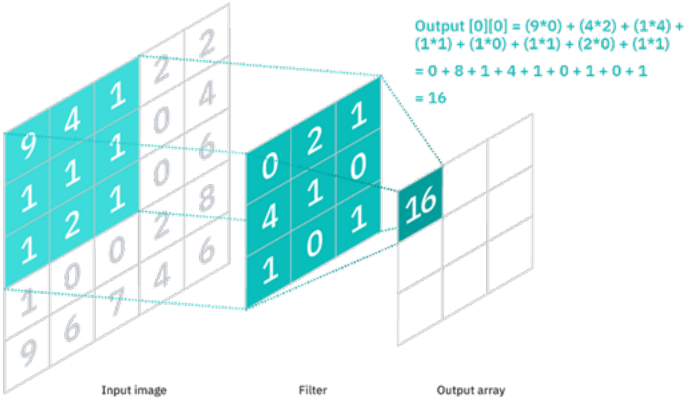


Figure 2.1: Convolution Layer

forms the basis of any convolutional neural network. In convolution operation, the arrays are multiplied element-wise, and the product is summed to create a new array. A convolution neural network has multiple hidden layers that help in extracting information from the data. The most important layers in CNN are Convolution Layer, ReLU Layer, Pooling Layer and Fully Connected Layer.

A convolution layer has several filters that perform the convolution operation. Every image is considered as a matrix of pixel values. The filter matrix is slides over the image and computes the dot product to get the convolved feature matrix.

ReLU stands for the rectified linear unit. Once the feature maps are extracted, the next step is to move them to a ReLU layer. ReLU performs an element-wise operation and sets all the negative pixels to 0. It introduces non-linearity to the network, and the generated output is a rectified feature map.

Pooling is a down-sampling operation that reduces the dimensionality of the feature map.

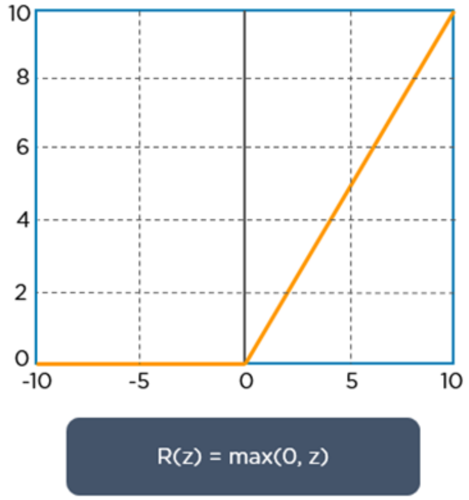


Figure 2.2: ReLU Layer

The rectified feature map now goes through a pooling layer to generate a pooled feature map. Example of max pooling is shown below.

The pooling layer uses various filters to identify different parts of the data like edges, corners, body, feathers, eyes, and beak etc.

The feature map generated after pooling operation is then given to a fully connected layer or dense layer for classification. In the dense layer, a sigmoid non-linearity activation function is applied over the features for predicting the output class labels.

2.2.2 RNN - Recurrent Neural Networks

Recurrent Neural Networks are extensions of feed-forward networks. In RNN, the output from one state is taken back as an input to it through a loop structure. RNN handles data in a sequential form and holds information in the network as a short memory through feedback loop. Thus, in a given time instant, the hidden state holds information from all its previous

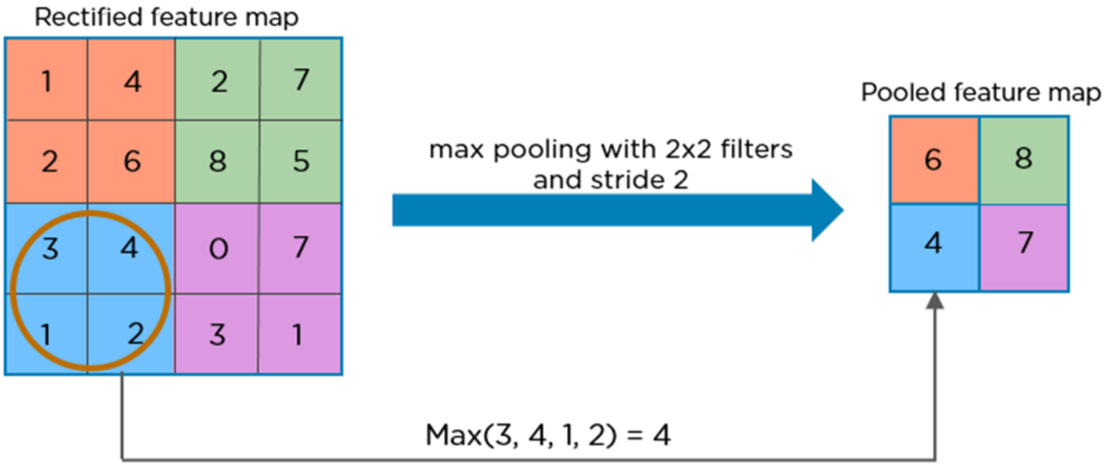


Figure 2.3: Pooling Layer

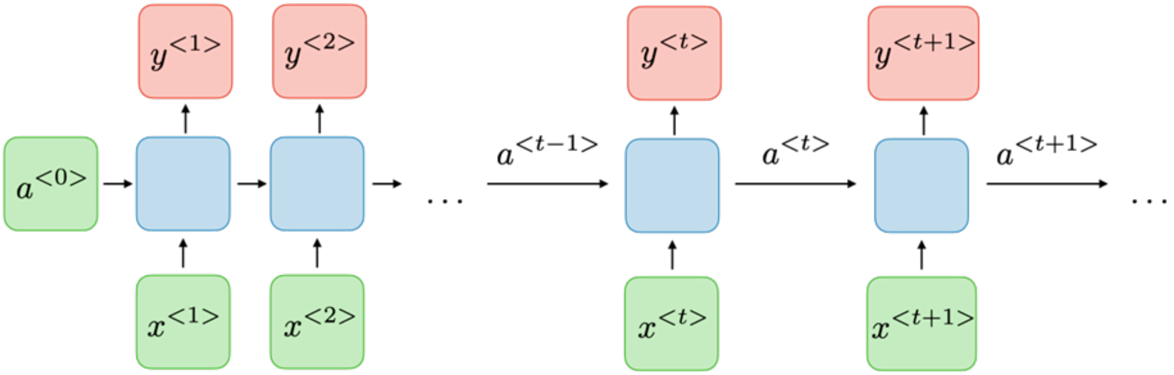


Figure 2.4: Working of Recurrent Neural Network

states. The feedback loop makes RNN as extensions of feed-forward network which passes the information only in one direction. The hidden states act as memory units in the network, receives input from previous states and transfers output to next state.

Backpropagation Through Time (BPTT) is a technique used in recurrent neural networks (RNNs) to compute gradients for the parameters of the network.

In a standard neural network, the input and output are assumed to be independent of each other, and the network processes the input in a feedforward manner. However, in an RNN, the output at each time step depends not only on the input at that time step but also on the internal state of the network (which is a function of the previous inputs and internal states). This creates a feedback loop, making it challenging to compute gradients for the parameters of the network.

BPTT is a way to compute gradients for the parameters of an RNN by ”unrolling” the network through time and treating it as a deep feedforward network. In this unrolled network, each time step is treated as a separate layer, and the parameters are shared across all time steps. The backpropagation algorithm is then used to compute the gradients at each time step, and these gradients are accumulated across time steps to update the parameters of the network.

However, the main drawback of BPTT is that it suffers from the vanishing gradient problem, where the gradients become very small as they are backpropagated through time, making it difficult for the network to learn long-term dependencies. This problem can be mitigated by using variants of RNNs such as LSTMs and GRUs that are specifically designed to handle long-term dependencies.

The main obstacle for RNN is the vanishing or exploding gradient problem, where the gradient vector explodes or decay over time steps. Hence, RNN fails to capture long-term

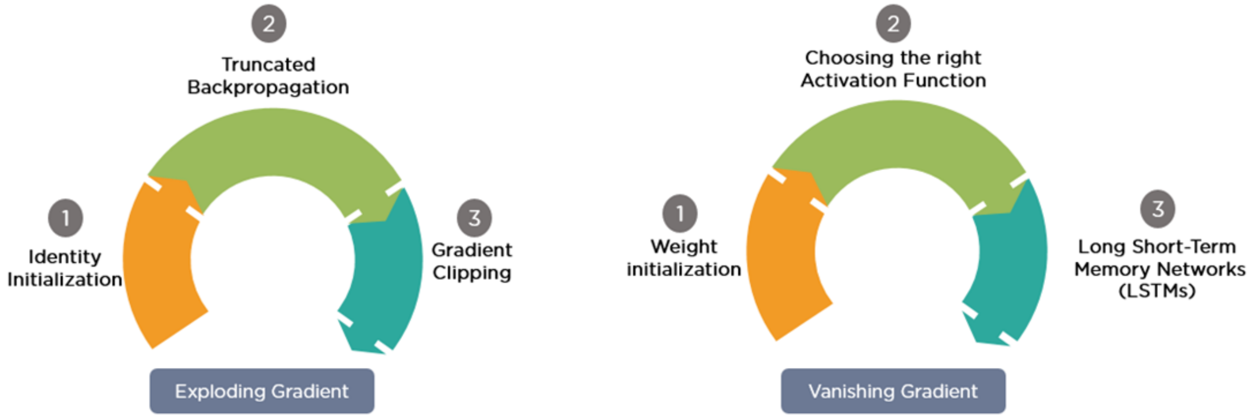


Figure 2.5: Exploding and Vanishing Gradient of Recurrent Neural Network

dependencies. Thus, we use LSTM and GRU for capturing long-term dependencies.

2.2.3 GRU - Gated Recurrent Unit

Gated recurrent unit are a gating mechanism in the recurrent neural networks. They are very similar to the LSTM units. The GRU comprises of the reset gate and the update gate instead of the input, output and the forget of the LSTM. It has fewer parameters than LSTM and it also lacks an output gate. There are several variations on the fully gated unit, with gating done using the previous hidden state and the bias in various combinations, and a simplified form called minimal gated unit.

2.2.4 LSTM - Long Short Term Memory

The fact that each LSTM cell has a mechanism involved contributes to the popularity of LSTM. In a typical RNN cell, the activation layer transforms the input at the timestamp and

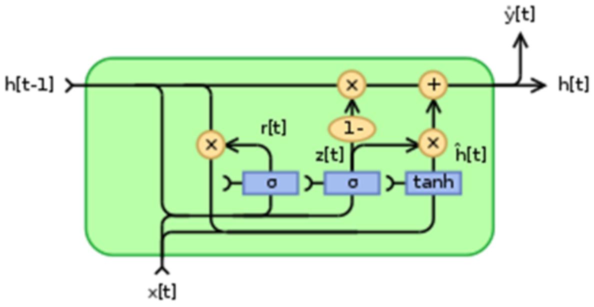


Figure 2.6: Fully Gated Recurrent Unit

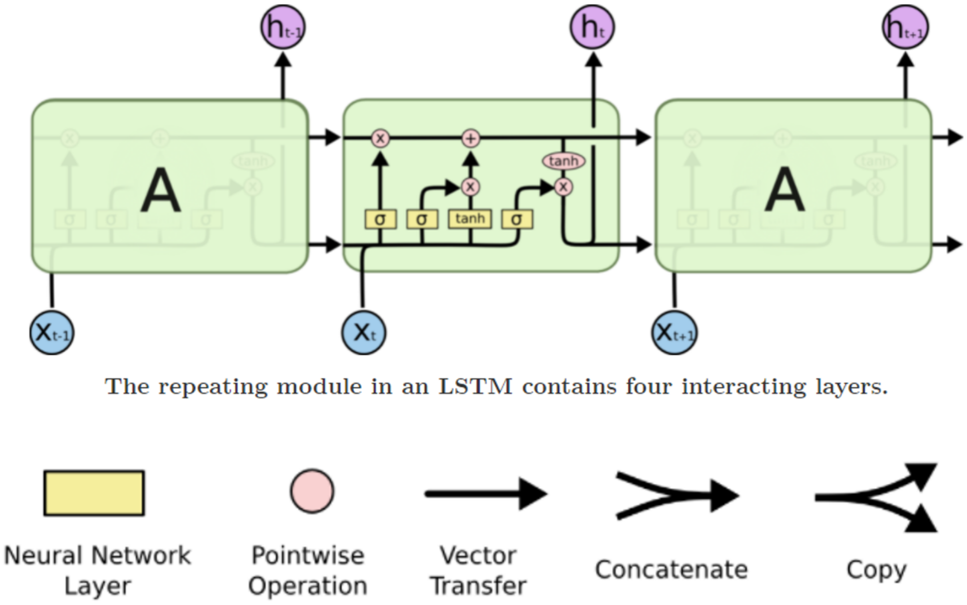


Figure 2.7: Single LSTM Unit

the hidden state from the previous time step into a new state. In contrast, the LSTM process is a little more complicated since it requires input from three separate states at once: the current input state, the short-term memory from the previous cell, and finally the long-term memory.

Before passing on the long-term and short-term information to the following cell, these cells employ the gates to control the information that will be maintained or deleted during loop operation. These gates can be thought of as filters that exclude undesired, chosen, and irrelevant information. The three gates that the LSTM employs are the input gate, forget gate, and output gate.

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.

The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.

The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!”.

An LSTM has three of these gates, to protect and control the cell state.

2.2.5 Inception

For accomplishing the video classification task, we used CNN-RNN based approach where CNN based models are used to extract features for each frame with the help of pretrained weights. One of the CNN based models we used for this task is Inception-v3. Inception-v3 is a convolutional neural network architecture that helps in learning the complex structures by using various filter sizes. he Inception CNN is designed to address the trade-off between depth and computational efficiency in standard CNNs by introducing several inception modules within the network. The inception module is a building block that performs multiple convolutions with different filter sizes in parallel, followed by a pooling operation, and then concatenates the resulting feature maps. This approach allows the network to capture both local and global features at multiple scales, while also reducing the number of parameters and computation required compared to traditional CNNs. The Inception CNN architecture consists of multiple stages, with each stage composed of several inception modules followed by a pooling operation. The number of filters and the size of the filters used in each module are optimized through network architecture search to maximize accuracy while minimizing computational cost. The Inception CNN architecture has achieved state-of-the-art performance

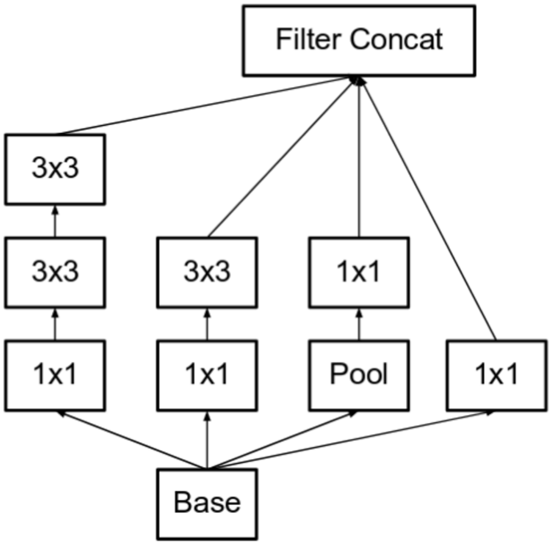


Figure 2.8: Single Cell of Inception

on several computer vision tasks, including image classification, object detection, and image segmentation.Using the ImageNet pretrained weights, we extract the features for each frame. After feature extraction is done, it is passed into RNN variants for classification as ragged tensors.

2.2.6 Xception

Another type CNN based model that we used for this task is Xception Net. The name ”Xception” is short for ”Extreme Inception”, as the architecture is based on the Inception CNN architecture but takes it to the extreme by replacing the standard inception module with a modified version called the ”depthwise separable convolution”. In standard CNNs, convolutional layers perform convolutions on all the input channels using a set of filters, followed by a pointwise convolution that combines the resulting feature maps into a smaller number of channels. In contrast, the depthwise separable convolution in Xception first performs a depthwise convolution that applies a separate filter to each input channel, followed by a pointwise convolution that combines the resulting feature maps into a smaller number of channels. This approach greatly reduces the number of parameters and computations required compared to standard convolutions, while still maintaining or even improving accuracy. The Xception CNN architecture consists of a series of convolutional blocks, each of which consists of a depthwise separable convolution followed by a batch normalization and activation function. The blocks are arranged in a hierarchical fashion to form a deep network that can learn increasingly complex features. The Xception architecture has achieved state-of-the-art performance on several computer vision tasks, including image classification, object detection, and semantic segmentation. The architecture has also been shown to be efficient and scalable, making it well-suited for deployment on mobile and embedded devices. The Xception convolutional neural network is a deep learning architecture that uses depthwise separable convolutions to reduce the number of parameters and computations while maintaining or improving accuracy, and has achieved state-of-the-art performance on various computer vision tasks. Xception Net has outperformed Inception net on ImageNet dataset and for our work Xception net with ImageNet weights is used.

2.2.7 Transformers

Transformers are a type of neural network architecture that have revolutionized the field of Natural Language Processing (NLP) in recent years. Unlike traditional recurrent neural

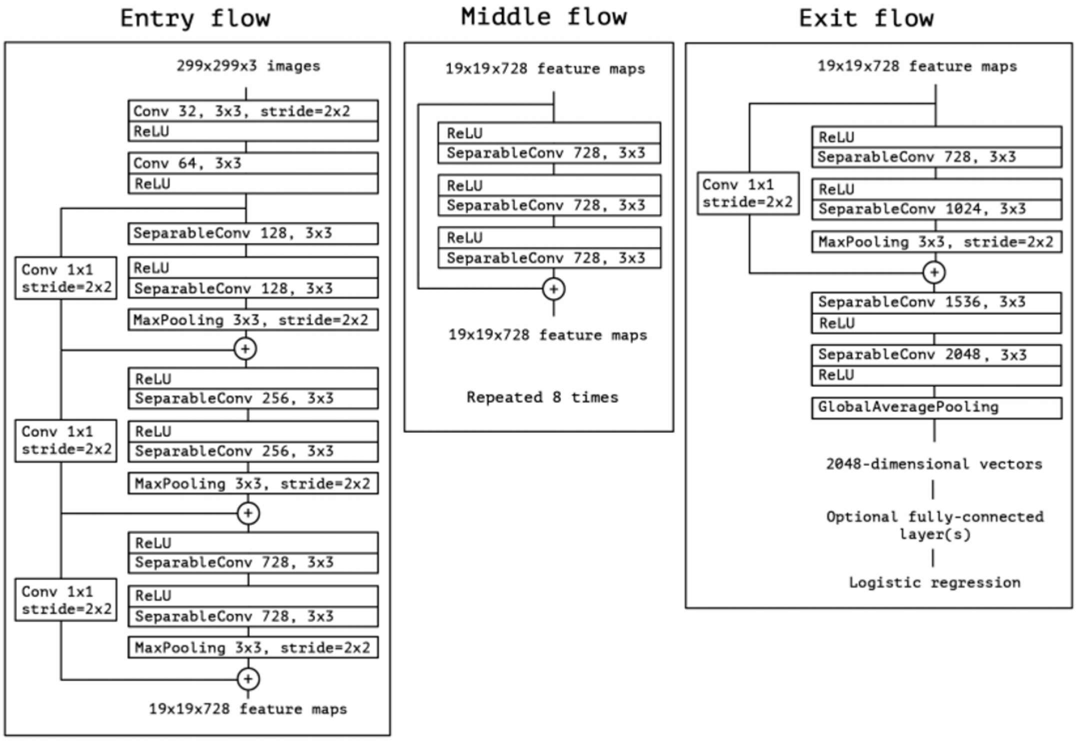


Figure 2.9: Single Cell of Xception

networks (RNNs) and convolutional neural networks (CNNs), transformers do not rely on sequential processing of text and instead process the entire input sentence at once. This parallel processing makes transformers much faster and more efficient than traditional architectures, and has enabled the development of powerful pre-trained language models that can be fine-tuned for a wide range of NLP tasks.

The key innovation behind transformers is the attention mechanism, which allows the model to selectively focus on different parts of the input sentence during processing. The attention mechanism computes a weighted sum of the input tokens, where the weights are learned during training and capture the importance of each token for the task at hand. By allowing the model to selectively attend to different parts of the input sentence, attention-based models are able to capture complex relationships between words and generate more accurate representations of the input text.

One of the most popular transformer-based models in NLP is the BERT (Bidirectional Encoder Representations from Transformers) model, which was introduced by Google in 2018. BERT is a pre-trained language model that is trained on massive amounts of text and can be fine-tuned for a wide range of downstream NLP tasks such as sentiment analysis, named entity recognition, and text classification. Since the introduction of BERT, several other transformer-based models have been introduced, including GPT (Generative Pre-trained Transformer) and T5 (Text-to-Text Transfer Transformer), each with their own unique strengths and applications. We used various transformer models such as MuRIL and Tamillion.

SimpleTransformers and Sentence Transformers are both popular Python libraries for working with transformer-based models in NLP. While both libraries are based on transformer-based models and provide a high-level API for building and training models, they have different focuses and use cases. SimpleTransformers is more general-purpose and can be used for a wide range of NLP tasks, while Sentence Transformers is more specialized and focused on sentence-level tasks. Ultimately, the choice between the two libraries will depend on the specific requirements of the project at hand.

2.2.7.1 Simple Transformers

SimpleTransformers is a high-level library for building and training transformer-based models, such as BERT and RoBERTa, for a variety of NLP tasks. The library provides a simple API that abstracts away many of the details of model building and training, making it easy for beginners to get started with transformer-based models. SimpleTransformers also includes several built-in features for handling tasks such as sequence classification, token classification, and question answering.

2.2.7.2 Sentence Transformers

Sentence Transformers, on the other hand, is a library specifically designed for building and training transformer-based models for sentence-level NLP tasks such as semantic textual similarity and sentence classification. The library provides pre-trained models, including BERT and RoBERTa, that can be fine-tuned on specific tasks using a simple API.

Chapter 3

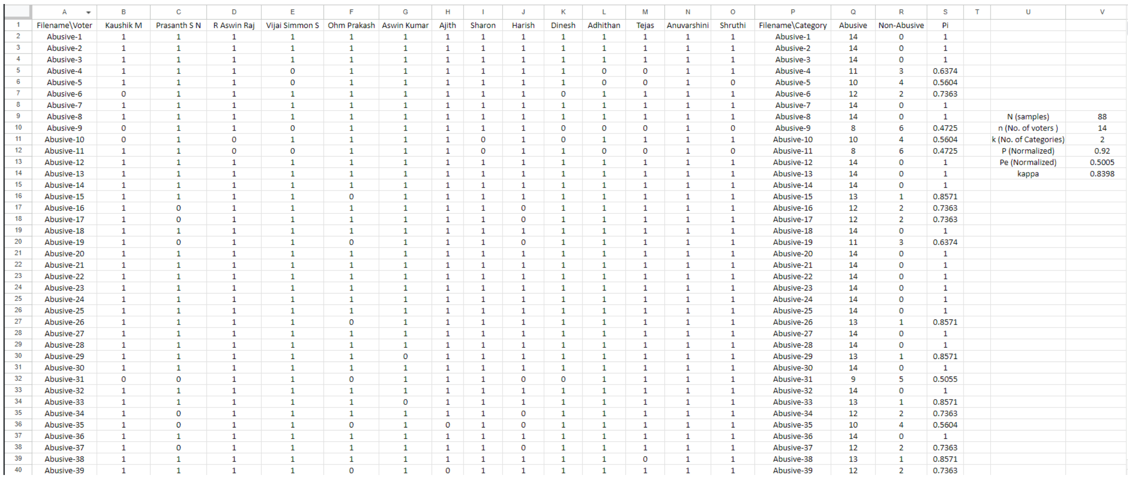
Dataset Description

This chapter describes about the dataset used for this work and how it was created and the hurdles we have faced in creating this dataset.

In order to develop a classification model, we need annotated data that has been annotated by experts or by a person with domain knowledge. As far as this work is concerned, we have created our own dataset that is the collection of videos accumulated from YouTube and manually transcripted texts and extracted audio from the corresponding videos. The data set is separated into three sets i.e., train, validation and test and annotated into two classes: Abusive and Non-Abusive. The annotation was done by various voters and the level of agreement between the voters is calculated using Fleiss Kappa.

3.1 Fleiss Kappa

Fleiss Kappa is a statistical measure for assessing the reliability of agreement between a fixed number of raters when assigning categorical ratings to a number of items or classifying items.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Filename\Voter	Kaushik M	Prasanth S N	R Aswin Raj	Vijai Simmon S	Ohm Prakash	Aswin Kumar	Ajith	Sharon	Harish	Dinesh	Adhithan	Tejas	Anuvarshini	Shruthi	Filename\Category	Abusive	Non-Abusive	Pi			
2	Abusive-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-1	14	0	1			
3	Abusive-2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-2	14	0	1			
4	Abusive-3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-3	14	0	1			
5	Abusive-4	1	1	1	0	1	1	1	1	1	1	0	0	1	1	Abusive-4	11	3	0.6374			
6	Abusive-5	1	1	1	0	1	1	1	1	1	1	0	0	1	0	Abusive-5	10	4	0.5604			
7	Abusive-6	0	1	1	1	1	1	1	1	1	0	1	1	1	1	Abusive-6	12	2	0.7363			
8	Abusive-7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-7	14	0	1			
9	Abusive-8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-8	14	0	1		N (samples)	88
10	Abusive-9	0	1	1	0	1	1	1	1	1	0	0	0	1	0	Abusive-9	8	6	0.4725		n (No. of voters)	14
11	Abusive-10	0	1	0	1	1	1	1	0	1	0	1	1	1	1	Abusive-10	10	4	0.5604		k (No. of Categories)	2
12	Abusive-11	1	1	0	0	1	1	1	0	1	1	0	0	1	0	Abusive-11	8	6	0.4725		P (Normalized)	0.92
13	Abusive-12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-12	14	0	1		Pe (Normalized)	0.5005
14	Abusive-13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-13	14	0	1		kappa	0.8398
15	Abusive-14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-14	14	0	1			
16	Abusive-15	1	1	1	1	0	1	1	1	1	1	1	1	1	1	Abusive-15	13	1	0.8571			
17	Abusive-16	1	0	1	1	1	1	1	1	0	1	1	1	1	1	Abusive-16	12	2	0.7363			
18	Abusive-17	1	0	1	1	1	1	1	1	0	1	1	1	1	1	Abusive-17	12	2	0.7363			
19	Abusive-18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-18	14	0	1			
20	Abusive-19	1	0	1	1	0	1	1	1	0	1	1	1	1	1	Abusive-19	11	3	0.6374			
21	Abusive-20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-20	14	0	1			
22	Abusive-21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-21	14	0	1			
23	Abusive-22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-22	14	0	1			
24	Abusive-23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-23	14	0	1			
25	Abusive-24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-24	14	0	1			
26	Abusive-25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-25	14	0	1			
27	Abusive-26	1	1	1	1	0	1	1	1	1	1	1	1	1	1	Abusive-26	13	1	0.8571			
28	Abusive-27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-27	14	0	1			
29	Abusive-28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-28	14	0	1			
30	Abusive-29	1	1	1	1	1	0	1	1	1	1	1	1	1	1	Abusive-29	13	1	0.8571			
31	Abusive-30	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-30	14	0	1			
32	Abusive-31	0	0	1	1	0	1	1	1	0	0	1	1	1	1	Abusive-31	9	5	0.5055			
33	Abusive-32	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-32	14	0	1			
34	Abusive-33	1	1	1	1	1	0	1	1	1	1	1	1	1	1	Abusive-33	13	1	0.8571			
35	Abusive-34	1	0	1	1	1	1	1	1	0	1	1	1	1	1	Abusive-34	12	2	0.7363			
36	Abusive-35	1	0	1	1	0	1	0	1	0	1	1	1	1	1	Abusive-35	10	4	0.5604			
37	Abusive-36	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Abusive-36	14	0	1			
38	Abusive-37	1	0	1	1	1	1	1	1	0	1	1	1	1	1	Abusive-37	12	2	0.7363			
39	Abusive-38	1	1	1	1	1	1	1	1	1	1	1	0	1	1	Abusive-38	13	1	0.8571			
40	Abusive-39	1	1	1	1	0	1	0	1	1	1	1	1	1	1	Abusive-39	12	2	0.7363			

Figure 3.1: Fleiss Kappa on Excel

Agreement can be thought of as follows, if a fixed number of people assign numerical ratings to a number of items, then the kappa will give a measure for how consistent the ratings are. If the raters are in complete agreement, then k=1. If there is no agreement among the raters (other than what would be expected by chance) then k¡=0 [1]. Table 1 shows the range of k-values and its corresponding interpretation.

$$k = \frac{\overline{P} - \overline{P_e}}{1 - \overline{P_e}}$$

Condition	K	Interpretation
Subjective Example: Only for annotators on two classes. See Landis & Koch 1977	≤ 0	Poor Agreement
	0.01 - 0.20	Slight Agreement
	0.21 - 0.40	Fair Agreement
	0.41 - 0.60	Moderate Agreement
	0.61 - 0.80	Substantial Agreement
	0.81 - 1.00	Almost Perfect Agreement

Table 3.1: Fleiss Kappa Value

3.2 Dataset distribution

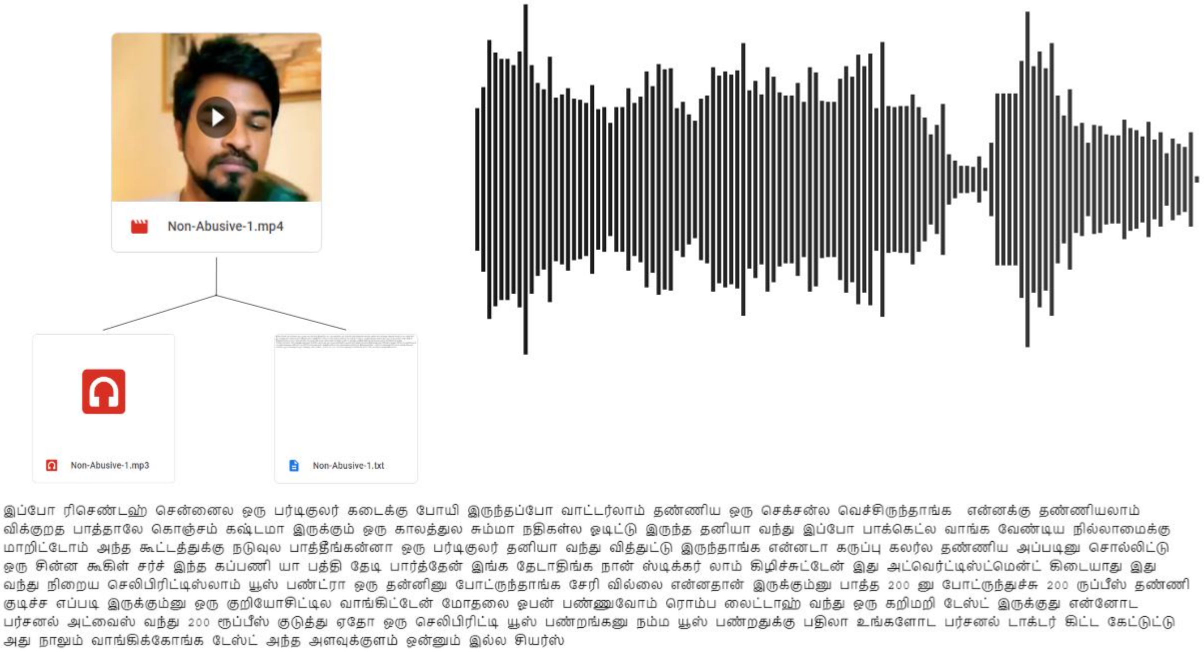
This work involves 7 voters whose mother tongue is Tamil and have knowledge about the data. The dataset consists of 88 videos, out of which 47 are abusive videos and 41 are non-abusive videos. The length of the videos is between 30 seconds to 1 minute. Table 2 shows the statistics of the dataset. This work is dealt with the Tamil Language dataset only. Textual elements were used in this work to complete the given job. To carry out the classification job, we used either uni-modal data (i.e., video, audio, or text) or multi-modal (i.e., a mix of any two or three modalities) characteristics. Each text is available as a *.txt file, audio as .mp3 file. Therefore, before beginning the experiment and assessment, we extracted all the texts and audios from the files.

Split	**Train**	**Test**	**Validation**
Percentage	80	10	10

Table 3.2: Dataset Split for train, test and validation

Parameters	**Value**
Frame rate of videos	900 per video
Video Format	.mp4
Sampling rate of speech signal	16k Hz
Average length of text data	731 characters
Maximum length of text data	1288 characters
Minimum length of text data	174 characters
Average length of video and audio	45 seconds
Maximum length of video and audio	60 seconds
Minimum length of video and audio	30 seconds

Table 3.3: Dataset Description



இப்போ ரிசெண்டஹ் சென்னைல ஒரு பர்டிகுலர் கடைக்கு போயி இருந்தப்போ வாட்டர்லாம் தண்ணிய ஒரு செக்சன்ல வெச்சிருந்தாங்க என்னக்கு தண்ணியலாம்
விக்குறத பாத்தாலே கொஞ்சம் கஷ்டமா இருக்கும் ஒரு காலத்துல சும்மா நதிகள்ல ஓடிட்டு இருந்த தனியா வந்து இப்போ பாக்கெட்ல வாங்க வேண்டிய நில்லாமைக்கு
மாறிட்டோம் அந்த கூட்டத்துக்கு நடுவுல பாத்தீங்கன்னா ஒரு பர்டிகுலர் தனியா வந்து வித்துட்டு இருந்தாங்க என்னடா கருப்பு கலர்ல தண்ணிய அப்படினு சொல்லிட்டு
ஒரு சின்ன கூகிள் சர்ச் இந்த கப்பணி யா பத்தி தேடி பார்த்தேன் இங்க தேடாதிங்க நான் ஸ்டிக்கர் லாம் கிழிச்சுட்டேன் இது அட்வெர்ட்டிஸ்ட்மென்ட் கிடையாது இது
வந்து நிறைய செலிபிரிட்டிஸ்லாம் யூஸ் பண்ட்ரா ஒரு தன்னினு போட்ருந்தாங்க சேரி வில்லை என்னதான் இருக்கும்னு பாத்த 200 னு போட்ருந்துச்சு 200 ருப்பீஸ் தண்ணி
குடிச்ச எப்படி இருக்கும்னு ஒரு குறியோசிட்டில வாங்கிட்டேன் மோதலை ஓபன் பண்ணுவோம் ரொம்ப லைட்டாஹ் வந்து ஒரு கறிமறி டேஸ்ட் இருக்குது என்னோட
பர்சனல் அட்வைஸ் வந்து 200 ரூப்பீஸ் குடுத்து ஏதோ ஒரு செலிபிரிட்டி யூஸ் பண்றங்கனு நம்ம யூஸ் பண்றதுக்கு பதிலா உங்களோட பர்சனல் டாக்டர் கிட்ட கேட்டுட்டு
அது நாலும் வாங்கிக்கோங்க டேஸ்ட் அந்த அளவுக்குளம் ஒன்னும் இல்ல சியர்ஸ்

Figure 3.2: Dataset Flow Chart

3.3 Dataset Creation

The acquisition of videos for developing a dataset on abusive and non-abusive behavior was an important and crucial task in our work. This dataset is crucial for developing algorithms and models that can accurately detect and classify abusive/non-abusive nature of the videos. The acquisition process involves collecting a diverse set of videos that depict both abusive and non-abusive behavior. These videos have been taken from the videos posted on Youtube by different vloggers. It is essential to ensure that the dataset is representative of the population and that it includes videos from different cultures, age groups, and genders.

Once the dataset is collected, it needs to be annotated and labeled to provide context and information about the behavior in the videos. The annotation process involves identifying and categorizing different types of abusive behavior, such as physical violence, verbal abuse, and sexual harassment.

Furthermore, a few more conditions were fixed to choose the videos, which are listed below,

- Length of the video

  In this work, the length of a video ranges from 30 seconds to 60 seconds. With a total average of video length of 45 seconds. This nonuniform duration of the videos is solely due to the fact that the dialogues take respective time to complete upon. Videos with more than a minute were trimmed down to avoid memory issues on lower spec machines.

- The face of the individual

  Videos were collected in such a way that the face of a reviewer is clearly visible. It helps to extract the required facial features from the data effectively. Therefore, videos with unclear faces and cropped faces were discarded from the selection of the dataset.

- Background of the video

  The background of the video was often selected with a fixed criteria in mind. The criteria is that the person who is speaking should only be present in the video without any other people or disturbances. Mostly the background of the videos is kept plain to avoid the extraction of visual features

3.4 Dataset Collection

This section describes challenges that we faced during the dataset collection part of our work:

- Acquisition of the videos

  The contents are published either in shorts format or full-length videos. The content present in the shorts format were easy to review and the nature of the video can be understood easily. But when it comes to full length videos, the content needs to be reviewed with atmost care to check for possible abusive content. The long format videos can be entirely abusive or partially abusive. Even though our dataset contains videos taken from YouTube, we had trimmed the videos to attract abusive or non – abusive stuff. Acquisition of abusive videos were been a tedious task due to the language and YouTube's policy constraint.

- Editing the video

  The duration of a video can range from 30 seconds to 20 minutes. The videos might contain a paid promotion or advertisement which is not required for work, since we are interested in talks that involve speech against an individual or a group or a product in either abusive or non-abusive nature. The video can contain both abusive and non-abusive speech where the speaker can speak about more than one object. So, it is essential for us to locate the beginning and end of such speech where only one particular object is being focused. We decided to keep the duration of the video to minimum of 30 seconds and maximum of 1 minute. If the speech exceeds more than 1 minute, we need to locate the end of a sentence to split the video such that the context and syntax is not lost. These processes are time-consuming as we have to spend time watching the video several times to edit the desired portion.

- Preparation of the transcripts

  The most challenging part of our work was to prepare the transcripts for our videos, which will be impossible to complete without the human assistance. We used speech-to-text and other text models for transcribing the videos. However, due to the pronunciation, improper sentence construction, less clarity in speech, dialect and pace of the speaker, the results were not satisfactory and needed human assistance to review the generated transcripts. The conditions were the models including the Google's speech-to-text model failed are,

  - Speaker may speak fast which made it difficult to perceive even for the native speakers.
  - Speaker stops the sentence abruptly. Due to this, placing the punctuations became another cumbersome task.
  - Obscured pronunciation by the speaker due to various reasons such as ignorance of the proper pronunciation, slip of the tongue, and native slang of the speaker.
  - To review and correct a video most efficiently, we had to watch the videos repeatedly and listen to the words one-by-one.
  - After the dataset collection, have to differentiate the data by naming abusive and non – abusive. Though we used Fleiss Kappa, we have to get the numerical ratings

to a number of items from fixed number of people who are native to the language Tamil to avoid the biased result.

The average time taken to complete a transcript for one video was 1 hour, making the process tedious and time-consuming

Chapter 4

Methodology

In the following subsections, we will see about the objective of our work and we will explain the proposed method in detail. The dataset consisting of abusive and non-abusive videos from the internet were collected. Each of these videos are of 30 seconds duration on average. The videos are first converted to mp3 format to obtain data for audio modality and from the audio file, as we are native tamil speakers, transcripted each of the audio file to create dataset for text modality.

For the text modality, we considered various feature extraction methods like count vectorizer, n-grams, tf-idf and n-grams character wise. We also considered transformers but it was quite complicated for us to consider it for multi modal approach.

4.1 Text Modality - Feature set

4.1.1 Count Vectorizer

CountVectorizer refers to the process of breaking down a phrase or any text into words by carrying out preprocessing operations like changing all words to lowercase and deleting special characters.

$$\text{CountVectorizer}(D)_{i,j} = \text{count}(t_j, d_i)$$

where:

'D' is the set of documents in the corpus

'i' is the index of the ith document in D

'j' is the index of the jth term in the vocabulary (i.e., the set of all unique terms in the corpus)

't_j' is the jth term in the vocabulary

'd_i' is the ith document in the corpus

count(t_j, d_i) is the number of times the jth term in the vocabulary (t_j) appears in the ith document (d_i)

4.1.2 n-grams

N-grams are continuous collections of objects from a corpus of text or voice, or nearly any kind of data. The number n in n-grams designates the size of the group of things to be taken into consideration; for example, a unigram for n = 1, a bigram for n = 2, a trigram for n =

3, and so on.

$$\text{n-grams}(w, n) = \{w_i, w_{i+1}, \ldots, w_{i+n-1}\}$$

where:

w is a sequence of words or tokens

n is the size of the n-grams

w_i is the ith word or token in the sequence w

4.1.3 TF-IDF

A popular statistical technique for information retrieval and natural language processing is called Term Frequency - Inverse Document Frequency (TF-IDF). It evaluates a term's significance inside a document in relation to a group of documents.

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

$$IDF(t, D) = \log\left[\frac{(|D| + 1)}{(df(t, D) + 1)}\right]$$

where:

t is the term (or word) for which you're calculating the score d is the document in which the term appears

D is the set of all documents in the corpus

TF(t,d) is the raw term frequency of term t in document d (i.e., the number of times term t appears in document d)

IDF(t,D) is the inverse document frequency of term t

4.1.4 n-grams character wise

Text documents can be represented as a series of characters to find character wise n-grams within them. A model is then trained using the n-grams that were retrieved from the sequence.

4.2 Text Modality Analysis

Once all of these features were obtained from the transcripts, we tried to identify the feature that best describes the necessary information from the transcripts to detect abusive content using various machine learning algorithms like SVM, XGBoost, Stochastic Gradient, etc. From the results it turned out to be TF-IDF that best describes the needful information for the text modality and hence was taken as the main feature for our first Multi-Modal approach.

4.3 Audio Modality - Feature set

For the audio modality, we considered features like MFCC and STFT from LibRosa and feature sets from OpenSmile library.

4.3.1 LibRosa

Librosa is a strong Python library designed to interact with audio and do analysis on it. It is the beginning point for dealing with audio data at scale for a variety of applications, such as identifying a person's voice or extracting personal traits from an audio.

- The task of supervised learning is speech recognition. The audio signal will be the input for the speech recognition issue, and we must predict the text from the audio signal. Since there will be a lot of noise in the audio signal, we cannot input the raw audio signal into our model. It has been shown that utilising features extracted from the audio signal as input to the basic model would result in significantly higher performance than using the raw audio signal as input. The most popular method for removing characteristics from an audio stream is called Mel-frequency cepstral coefficients(MFCC).

- The frequency content of a non-stationary signal is examined using the short-time Fourier transform (STFT). The spectrogram time-frequency representation of the signal is defined as the magnitude squared of the STFT.

4.3.2 OpenSMILE

OpenSMILE is a toolset for extracting and categorising audio features from speech and music signals. It stands for "open-source Speech and Music Interpretation by Large-space Extraction." Widespread use of openSMILE is in affective computing's automated emotion recognition.

4.4 Audio Modality Analysis

After extracting features using the OpenSmile and LibRosa libraries, we utilized various machine learning algorithms to identify the best feature set and feature level pair towards the initial approach for Multi-Modal classification. From the results, features from OpenSmile with ComParE_2016 Feature set with Low Level Descriptor turned out to be the best out of the set. We then tried with Deep Learning models to determine the best feature that can standalone for the audio modality and it turns out that due to lower dimensions, ComParE_2016 Feature set with Low Level Descriptor can not be used for Residual Network and the feature set from LibRosa package gave the best results out of the set and therefore was considered as the only feature for audio modality for our initial approach towards Multi-Modal classification.

4.5 Video Modality Analysis

For the Video Modality, two approaches were considered - frames with padding and frames without padding. Average duration of each video is 45 seconds. For padding purpose, we considered 900 frames as the cap for all the videos. Whichever video had fewer number of frames, those videos get padded with frames value set to zero and those videos with frames count higher than 900 got trimmed off. Each of the videos were sent to three different deep learning models like convolutional neural networks, Gated recurrent unit (GRU) and Long-Short Term Memory (LSTM) model.

From the results obtained it was noted that the GRU model gave the best results while testing for video modality alone and therefore was considered for the multi modal approach.

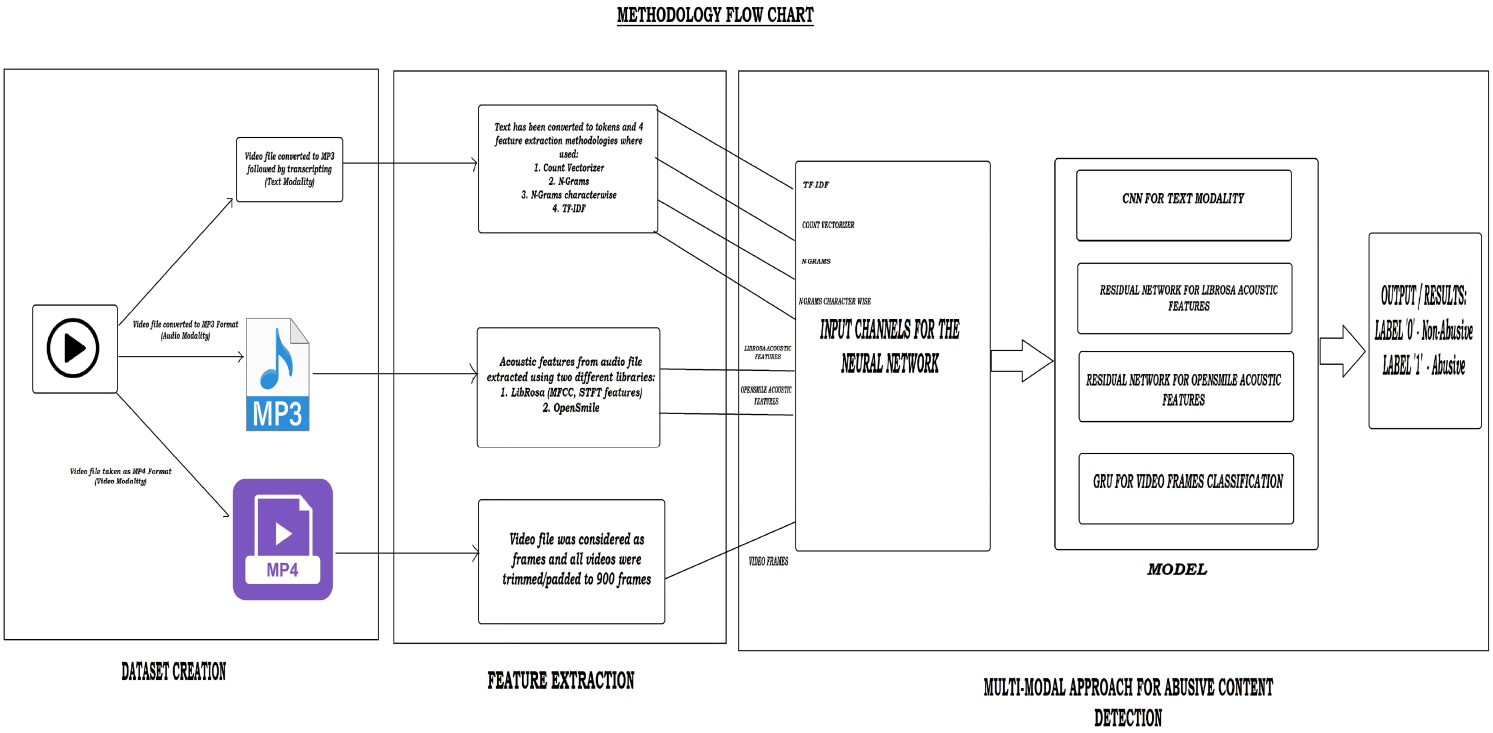

METHODOLOGY FLOW CHART
Video file converted to MP3 followed by transcripting (Text Modality)
Video file converted to MP3 Format (Audio Modality)
MP3
Video file taken as MP4 Format (Video Modality)
MP4
DATASET CREATION
Text has been converted to tokens and 4 feature extraction methodologies where used:
1. Count Vectorizer
2. N-Grams
3. N-Grams characterwise
4. TF-IDF
Acoustic features from audio file extracted using two different libraries:
1. LibRosa (MFCC, STFT features)
2. OpenSmile
Video file was considered as frames and all videos were trimmed/padded to 900 frames
FEATURE EXTRACTION
TF-IDF
COUNT VECTORIZER
N-GRAMS
N-GRAMS CHARACTER WISE
LIBROSA ACOUSTIC FEATURES
OPENSMILE ACOUSTIC FEATURES
VIDEO FRAMES
INPUT CHANNELS FOR THE NEURAL NETWORK
CNN FOR TEXT MODALITY
RESIDUAL NETWORK FOR LIBROSA ACOUSTIC FEATURES
RESIDUAL NETWORK FOR OPENSMILE ACOUSTIC FEATURES
GRU FOR VIDEO FRAMES CLASSIFICATION
MODEL
OUTPUT / RESULTS:
LABEL '0' - Non-Abusive
LABEL '1' - Abusive
MULTI-MODAL APPROACH FOR ABUSIVE CONTENT DETECTION

Figure 4.1: Methodology Flow Chart

4.6 MultiModal Analysis

Once these features are extracted and passed into corresponding neural network branches, they get encapsulated into a single model using concatenation followed by the output layer with sigmoid activation function since it is a binary classification. The model is then trained with binary cross entropy loss function and with the Adam learning rate optimizer value set to 0.001. The model is set with a cap on number of epochs to be 200 and with the help of early stopping we arrive with a model that is well trained.

Chapter 5

Result and Inferences

This chapter describes about our results that we obtained for each model with respect to each modality.

5.1 Audio Modality

Table 4-12 shows our results obtained on using audio modality for the classification. The best results among the ML models were obtained when Random Forest is used along with features extracted using OpenSMILE (ComParE_2016 with Low Level Descriptor). Among DL models, the best results were obtained when we used residual networks with LibRosa package.

5.2 Text Modality

Table 13-18 shows our results obtained on using text modality for the classification. The best results among the ML models were obtained when XGBoost is used along with features extracted using TF-IDF. Due to the time-complexity of

One-Hot encoding. When transformers were used to feature extraction and passed to ML models, they performed subpar whereas using simple transformers performed almost on par with models without using transformers.

5.3 Video Modality

Table 19 shows our results obtained on using video modality for the classification. We tried with and without padding for the videos. Among the CNN-RNN and its variant models, the CNN-GRU model performed the best with videos trained without padding among all the combination of the models with padding/no-padding.

5.4 Multi-modal Approach

Table 20 shows our results obtained on using Multi Modal approach for the classification. Multi Modal approach involving audio and text performed better than the approach involving all the three modalities. Among the approaches involving all the three modalities, model with higher number channels performed much better when compared to single channel model.

Models	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression {Penalty:l2, Solver:lbfgs}	0.61	0.5	0.5	0.48
SVM {C:1, Kernel:RBF}	0.67	0.59	0.54	0.52
Random Forest {Criterion:Gini, n estimator:100}	**0.83**	**0.83**	**0.88**	**0.83**

Table 5.1: OpenSmile (FeatureSet: ComParE_2016, FeatureLevel : Functionals)

Models	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression {Penalty:l2, Solver:lbfgs}	0.61	0.66	0.67	0.61
SVM {C:1, Kernel:RBF}	**0.67**	**0.75**	**0.75**	**0.67**
Random Forest {Criterion:Gini, n estimator:100}	0.67	0.69	0.71	0.66

Table 5.2: OpenSmile (FeatureSet: eGeMAPSv02, FeatureLevel: Functionals)

Models	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression {Penalty:l2, Solver:lbfgs}	0.78	0.80	0.83	0.77
SVM {C:1, Kernel:RBF}	0.61	0.73	0.71	0.61
Random Forest {Criterion:Gini, n estimator:100}	**0.94**	**0.93**	**0.96**	**0.94**

Table 5.3: OpenSmile (FeatureSet: ComParE2016, FeatureLevel : LowLevelDescriptor)

Models	**Accuracy Normal**	**Accuracy Scaled**	**Accuracy MinMax**	**Test Normal**	**Test Scaled**	**Test MinMax**
K Neighbors Classifier	0.72	0.50	0.50	0.72	0.50	0.50
SVC	0.78	0.61	0.61	0.78	0.61	0.61
SVC-RBF Kernel	0.78	0.44	0.61	0.78	0.44	0.61
Decision Tree Classifier	0.61	0.28	0.50	0.61	0.28	0.50
Random Forest Classifier	0.72	0.39	0.50	0.72	0.39	0.50
Logistic Regression	0.72	0.39	0.50	0.72	0.39	0.50

Table 5.4: LibRosa(FeatureSet:MFCC, mel spectrogram, STFT)

Models	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Baseline Model Network	0.55	0.73	0.63	0.53
Simple U Network Model	**0.77**	**0.76**	**0.76**	**0.77**
Residual Network	NA	NA	NA	NA

Table 5.5: OpenSmile (FeatureSet: ComParE2016, FeatureLevel : LowLevelDescriptor)

Models	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Baseline Model Network	0.50	0.73	0.55	0.41
Simple U Network Model	**0.61**	**0.79**	**0.56**	**0.51**

Table 5.6: OpenSmile (FeatureSet: ComParE_2016, FeatureLevel : Functionals)

Models	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Baseline Model Network	0.55	0.58	0.57	0.55
Simple U Network Model	**0.61**	**0.60**	**0.60**	**0.61**
Residual Network	NA	NA	NA	NA

Table 5.7: OpenSmile (FeatureSet: eGeMAPSv02, FeatureLevel: Functionals)

Models	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Baseline Model Network	0.77	0.75	0.85	0.75
Simple U Network Model	0.77	0.75	0.85	0.79
Residual Network	**0.88**	**0.83**	**0.92**	**0.89**

Table 5.8: LibRosa (FeatureSet:MFCC, mel spectrogram, STFT

Feature	**Accuracy**	**Precision**	**Recall**	**F1 Score**
One Hot Encoding	**0.94**	0.947	**1.0**	**1.0**
TF IDF	**0.94**	**0.952**	**1.0**	**1.0**
TF IDF N Grams	0.83	0.87	0.99	0.988
TF IDF N Grams Characterwise	textbf0.94	**0.952**	0.968	0.963
LaBSE	**0.94**	0.94	0.95	0.94
Distiluse v1	0.56	0.36	0.28	0.50
Tamilion	0.56	0.36	0.28	0.50
MuRIL	0.56	0.36	0.28	0.50

Table 5.9: Text Modality: Logistic Regression

Feature	**Accuracy**	**Precision**	**Recall**	**F1 Score**
One Hot Encoding	0.88	0.889	**0.956**	0.9
TF IDF	**0.94**	**0.952**	0.955	**0.938**
TF IDF N Grams	0.88	0.87	0.938	0.919
TF IDF N Grams Characterwise	0.83	0.842	0.944	0.887
LaBSE	0.72	0.72	0.75	0.74
Distiluse v1	0.50	0.41	0.74	0.55
Tamilion	0.44	0.31	0.22	0.50
MuRIL	0.56	0.36	0.28	0.50

Table 5.10: Text Modality: Stochastic Gradient

Feature	**Accuracy**	**Precision**	**Recall**	**F1 Score**
One Hot Encoding	0.88	0.909	0.917	0.875
TF IDF	**0.94**	**0.952**	0.955	**0.938**
TF IDF N Grams	0.83	0.87	0.885	0.812
TF IDF N Grams Characterwise	**0.94**	**0.952**	0.955	**0.938**
LaBSE	0.78	0.78	0.79	0.79
Distiluse v1	0.78	0.78	0.78	0.78
Tamilion	0.44	0.31	0.22	0.50
MuRIL	0.56	0.36	0.28	0.50

Table 5.11: Text Modality: SVM

Feature	**Accuracy**	**Precision**	**Recall**	**F1 Score**
One Hot Encoding	**0.94**	**0.952**	0.955	0.950
TF IDF	0.83	0.857	0.916	0.906
TF IDF N Grams	0.88	0.909	**0.983**	**0.975**
TF IDF N Grams Characterwise	0.88	0.90	0.962	0.956
LaBSE	0.83	0.83	0.83	0.84
Distiluse v1	0.83	0.83	0.83	0.84
Tamilion	0.56	0.36	0.28	0.50
MuRIL	0.89	0.89	0.89	0.89

Table 5.12: Text Modality: Random Forest

Feature	**Accuracy**	**Precision**	**Recall**	**F1 Score**
One Hot Encoding	0.94	0.952	**1.0**	**1.0**
TF IDF	**1.0**	**1.0**	**1.0**	**1.0**
TF IDF N Grams	0.83	0.842	0.875	0.875
TF IDF N Grams Characterwise	0.88	0.90	0.976	0.963
LaBSE	0.78	0.78	0.79	0.79
Distiluse v1	0.78	0.78	0.78	0.78
Tamilion	0.50	0.49	0.49	0.49
MuRIL	0.89	0.89	0.90	0.90

Table 5.13: Text Modality: XGBoost

Feature	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Tamilion	**0.94**	**0.94**	**0.95**	**0.95**
MuRIL	0.83	0.83	0.83	0.88

Table 5.14: Text Modality: Simple Transformers

Models	**Accuracy With Padding**	**Accuracy Without Padding**
CNN-RNN	0.87	0.78
CNN-GRU	0.72	**1.0**
CNN-LSTM	**0.89**	0.67

Table 5.15: Video Modality

Models	**Training Accuracy**	**Validation Accuracy**
MultiModal - Audio + Text	0.88	0.88
MultiModal - Audio + Text + Video (1 Channel per Modality)	0.57	0.77
MultiModal - Audio + Text + Video (Multiple Channel per Modality)	**0.92**	**0.83**

Table 5.16: Multi Modality

Chapter 6

Discussion

Our initial approach of uni-modal classification gave us the aspect of understanding our dataset. The text modality analysis with multiple machine learning models and transformers led us to considering TF-IDF as the feature that captures necessary information from the dataset compared to other features. The audio modality analysis had two different approaches - features extracted using OpenSmile library and from LibRosa library. Both had its own merits and demerits but the feature set extracted from LibRosa librarycame on top given that the feature set dimension from OpenSmile package are not enough to be passed through residual network. For the video modality, the video frames were padded to 900 frames per video (which means 30 frames per second) and were analysed using different convolutional neural networks and Gated Recurrent Unit gave the best results. After analyzing the uni-modal approaches, we jotted down to the Multi-modal approach.

The first Multi-Modal model was not as quite effective as we expected it to perform. For text modality, TF-IDF was the feature set considered. For audio modality, the features like MFCC and STFT from LibRosa library were considered and the GRU network for the padded video frames were considered. This approach failed as the performance was way below par when compared to the uni-modal approaches of these features.

The second Multi-Modal approach with multiple feature sets for each of the modality turned out to be a fruitful one which enhanced the performance of the model compared to the first approach of having only single feature for each of the modality. For Text modality, the features like Count Vectorizer, TF-IDF, TF-IDF N grams and TF-IDF N grams characterwise were used. For the audio modality, the feature sets from both the libraries - OpenSmile and LibRosa were used. For the video modality, the videos with padded frames were considered. The presented model is a robust one as it considers various features for each of the modality and will turn out to be an effective and scalable solution to the moden era - identifying the Abusive content in Tamil language that is spread across various social media platforms.

The proposed model needs any video obtained from the social media platforms, which has to be trimmed to windows of 30 seconds and should be converted to MP3 format. The automated work for the same is attached with the project. The success of our Multi-Modal approach conveys that the hate and abusive content across any social media platform can be eliminated with recurrent training to our model. The main findings of this project reveals that the collective information that is obtained from different modality gives promising results when compared to analyzing each of the modalities in a singular manner and that is how social media platforms like Facebook and Twitter are able to eliminate abusive content for a selective languages. Our model is ready to deploy that helps the society eliminate the racial, hateful and abusive usage of Tamil as a language.

6.1 Limitations and Countermeasures

In our research, although the proposed Multi-Modal model showed successful abusive content detection and better performance, they are limited to some factors like data set creation. For both training and testing purpose, it is a tedious task to create the data set. It is a smooth process till we trim the video to a 30 seconds window and converting them to audio file. The problem arises when we try automating the transcription process by passing audio file to applications like the Google API. As these applications are not trained with abusive words, the transcription of abusive content gives bizarre outputs and human interference is required to convert the audio files to text transcripts. The person has to understand Tamil language and better be a Native Tamil language speaker in order to obtain accurate transcriptions. This is the only hurdle in utilizing and deploying our model.

Chapter 7

Conclusion and future work

As a part of our future works, we like to improve our work in such a way that, instead of extracting audio and transcript manually, we will try to automate this process via API that are already present in the literature. We also like to investigate about the future improvements and developments towards this ideology. We will also try to do a real-world implementation based on which our project will get optimized in successive updates. .

Bibliography

[1] Sarah A. Abdu, Ahmed H. Yousef, and Ashraf Salem. “Multimodal Video Sentiment Analysis Using Deep Learning Approaches, a Survey”. In: *Information Fusion* 76 (2021), pp. 204–226. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2021.06.003. URL: https://www.sciencedirect.com/science/article/pii/S1566253521001299.

[2] Bharathi Raja Chakravarthi et al. *DravidianMultiModality: A Dataset for Multimodal Sentiment Analysis in Tamil and Malayalam*. 2021. arXiv: 2106.04853 [cs.CL].

[3] Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. *Towards multidomain and multilingual abusive language detection: a survey*. https://link.springer.com/article/10.1007/s00779-021-01609-1. Aug. 2021.

[4] Prasanth S N et al. “CEN-Tamil@DravidianLangTech-ACL2022: Abusive Comment detection in Tamil using TF-IDF and Random Kitchen Sink Algorithm”. In: *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 70–74. DOI: 10.18653/v1/2022.dravidianlangtech-1.11. URL: https://aclanthology.org/2022.dravidianlangtech-1.11.

[5] Rini Sharon et al. *Multilingual and Multimodal Abuse Detection*. 2022. arXiv: 2204.02263 [eess.AS].