Hate Me Not: Detecting Hate Inducing Memes in Code Switched Languages

Kshitij Rajput*
Netaji Subhas University of Technology
Delhi, India
rajput.kshitij97@gmail.com

Raghav Kapoor*
Netaji Subhas University of Technology
Delhi, India
raghavk.co@nsit.net.in

Kaushal Kumar Rai*
Netaji Subhas University of Technology
Delhi, India
kaushalk.co@nsit.net.in

Preeti Kaur
Netaji Subhas University of Technology
Delhi, India
preetikaurnsit@gmail.com

ABSTRACT

With the advent of social media, the rise in the number of social media users has upsurged the hateful content being posted online. In countries like India, where multiple languages are spoken, these abhorrent posts are not only from a particular language but rather from an unusual blend of languages called code-switched languages. In India, a particular pair, namely *Hinglish* is the most popular and the task of hate speech detection gains its intricacy from the fact that there are no fixed spellings, grammar and semantics for this language. Also, this hate speech is depicted with the help of images to form *"Memes"* which create a long-lasting impact on the human mind. This poses a substantial threat for the users that often become a victim of the derogatory speech and abuses on such platforms. In this paper, we take up the task of hate and offense detection from multimodal data, i.e. images (Memes) that contain text in code-switched languages. We firstly present a novel triply annotated Indian political Memes (IPM) dataset, which comprises memes from various Indian political events that have taken place post-independence and are classified into three distinct categories. We also propose a binary-channeled CNN cum LSTM based model wherein we individually process the images using the CNN model and text extracted from the images using the LSTM model and finally recombine them to get the result. This one-of-a-kind model outperforms all other models to give state-of-the-art results in the domain of offensive image (Memes) classification in code-switched languages - *Hinglish.* We also release the code, model and IPM dataset for research purposes.

CCS CONCEPTS

• **Information systems → Specialized information retrieval**; • **Computing methodologies → Natural language processing**; *Supervised learning by regression*; *Neural networks*;

KEYWORDS

Web mining and content analysis, Neural Networks, Hate Speech Detection, Multimodal, Datasets, Code Switching, Indian Political Memes Classification, Social Media Analysis

1 INTRODUCTION

During a short period, social media has witnessed a massive growth of users who engage with various social media platforms to express their emotions either through textual comments or through the use of images and videos. Over time people have extensively started using multimodal means to express their sentiments as they believe it is a much stronger way to depict the accurate sense as to how they feel about a particular person or a situation. According to an article [1], Facebook status updates with images get 2.3 times more engagement than status updates without images. It is much easier for a human to process the visual information and about 90% of all information that we perceive and that gets transmitted to our brains is visual [2]. A recent study concluded that the loneliness of a social media user can be reduced through image-based social media usage rather than textual use of social media [24] which clearly states how the images have become an integral part of expressing our emotions.

Some of the primary purposes for which such content is posted online are personal branding, humor (*Memes*), digital marketing, political canvassing. A large proportion of these images give rise to the problem of hate speech. With the growing popularity of social media; hate-inducing and violent content are growing disproportionately and this is specifically during the time of elections or any such political event. In many cases, this hate is expressed in Memes by the means of *sarcasm*. Users often club these images with text to convey their angst. Hate speech detection in such imageries is an intricate task as it involves the analysis of images as well as of the text within it.

The problem of hate speech has been rising around the world, specifically in India, and according to the various social networks and government policies, one is not allowed to misuse the right to speech to provoke other religions or communities [26]. Hate speech detection in Indian languages is a complex problem due to its rich linguistic diversity [1]. The world of social media has also given rise to code-switched languages. In India, a particular pair of code mixed language, *Hinglish* is most popularly used. Hinglish language consists of non-fixed grammar, irregular semantics and spellings. The plethora of slang and profane words and randomized spelling variations is one of the few characteristics which make the

*These authors contributed equally to the paper

[1] https://bit.ly/3fCjGYr
[2] http://bit.ly/2Ht3ubm

task of hate speech detection much difficult in a pseudo-language (code-switched language) than for ordinary languages. In Hinglish language, the words are written in the Roman script instead of the *Devnagari* script. However, the meaning of the words is actually in Hindi. The same word can be written in many ways. For example, in the sentence "*ye bahut swaad hai*", the word *swaad*, which means tasty, can be written as *swad*, *swaad*, *svaad*, *svad* which all mean the same thing. Also, the word-to-word translation of the sentence would be "This very tasty is" which is grammatically incorrect in English. Hence, as we see, the task gains even more complexity when exhibited in code-switched languages. The vast number of social media users, the rise in hate speech, ambiguities in the semantics and grammar of Hinglish, a labyrinth of image analysis is what demonstrates the magnitude of the problem.

In the paper, we present a deep learning solution to solve all obstacles to classify the images (Memes) on social media in one of the three categories: *Hate Inducing*, *Satirical* and *Benign or Non-Offensive*. We extract the text from the images and process the text and the image independently, and finally combine the results to get the final category in which the complete image with text can be classified. Inspired by the works of [16], we use LSTM based model for the text classification. For text classification different word embedding models have been tried including Glove [23], FastText [4], Word2Vec [14], Bert [11] and embeddings. The images have been analyzed with the help of a CNN-based model.

With this in mind, a doubly annotated dataset consisting of Indian Political memes which are classified in three categories has been created and released along with the model.

The significant contributions in our work are as follows :

(i) Creation of IPM (Indian Political Memes) dataset.

(ii) Demonstrating how independent analysis of text and images and then recombining the results gives significantly better results than considering the image alone.

(iii) Creation of a deep learning-based classifier model which will outperform all the other baseline models on the IPM dataset.

In section 2, we discuss the background and related work that has been done in this field. This is followed by section 3 where we discuss the newly created dataset called the IPM dataset. In section 4, we will elaborately explain the methodology used which will be followed by the results. Finally, in sections 5 and 6 we briefly lay out the conclusion and the future enhancements that can be made to our work.

2 BACKGROUND AND RELATED WORK

The task of automatic detection of hate speech is fast becoming an important problem in today's world with the increasing penetration of the internet among users in the past decade. The initial task for hate speech detection was performed by [33], who developed a prototype system *Smokey* for detecting email flames (angry or offensive emails) using 47 elements feature set which captured the syntax and semantics of the sentences present in the dataset. These features were passed to a decision tree generator to categorize the emails as flames or not. Yin et al. [41] used libSVM as the classifier model with local features, sentiment features and contextual features for detecting harassment on Web 2.0. An SVM-based model trained on a corpus of 1,655,131 user comments on Yahoo buzz, combined with valence analysis for detecting personal insults on social news websites was put forward by [32]. Another work, [40] proposed to use Latent Dirichlet Allocation (LDA) for generating topical features which are passed to logistic regression (LR) classifier for the task of detecting offensive tweets on a Twitter corpus. Gamback et al. [13] proposed a Convolution Neural Network (CNN) for classification on Twitter text data into four categories: sexism, racism, both (racism and sexism) and non-hate-speech using the following features: character 4-grams, word2vec vectors to capture semantic information, randomly generated word vectors, and word vectors combined with character n-grams. A dataset of about 25K tweets labeled as hate-inducing, offensive or benign was released by [9] who put forward a logistic regression model with L2 regularization for classifying the tweets in one of the three categories.

However, most of the research on hate speech detection in the past was restricted to English text only. The task of hate speech detection on the Italian language using the following features : (i) morpho-syntactical features, (ii) sentiment polarity and (iii) word embedding lexicons was shown by [10]. Two types of classifiers were considered, first SVM based classifier and secondly LSTM based classifier which was compared for hate speech detection on the Italian tweets dataset. However, the task of hate speech detection on code-switched data has its own intricacies of having to deal with non-fixed spellings, grammar and semantics for this language.

Since our work consists of hate speech detection of memes with text in Hinglish language, so we look at some past work for hate speech detection focusing on Hinglish language. The task of hate speech detection on Hindi-English code switched data using a Random Forest (RF) classifier and a Support Vector Machine (SVM) classifier was performed by [3] using the following features: (i) character n-grams, (ii) word n-grams, (iii) punctuations, (iv) negation words and (v) lexicons. A ternary trans CNN model using transfer learning for hate speech detection on Hindi-English code switched dataset was performed by [20]. Further work was proposed by [16] on hate speech detection on Hindi-English code switched HEOT dataset using LSTM based model with transfer learning which takes glove embeddings and word2vec embeddings as features. Mathur et al. [19] also put forward a MIMCT model which takes in a series of primary and secondary word embeddings into a CNN-LSTM based binary channel neural architecture for hate speech detection.

Analyzing sentiments from an image is a complex problem in itself and has seen many works in the past. Research by [39] focused on analyzing sentiment out of images where the authors proposed a novel mechanism of finding the emotion out of an image by finding an orthogonal three-dimensional factor space of an image and then passing it through an SVM classifier. Siersdorfer et al. [31] analyzed the relation between the sentiment of images expressed in metadata and their visual content in the social photo-sharing environment Flickr. Another work by [18] deals with the training of a deep convolutional neural network to classify the 1.2 million in ImageNet into 1000 different categories. A progressive CNN model for visual sentiment analysis with transfer learning to learn the features on a Twitter image dataset was also proposed by [42].

Since our work focuses on combining both the textual and the image features from a meme, we look at some researches that have performed classification considering such a combination of

Table 1: Class Distribution in Indian Political Memes (IPM) dataset.

Label	IPM Dataset
Non-Offensive	339
Hate-inducing	427
Satirical	452
Total	**1218**

features before. Cai et al. [6] performed sentiment analysis on the combination of text and images instead of considering them separately. Two individual CNN structures were used to capture the image and textual features, which were then passed to another CNN structure to exploit these features and calculate the sentiment. A novel method was proposed by [25] for capturing textual and visual features using deep convolutional networks and then passing these features to a multiple kernel learning classifier for performing the task of multimodal sentiment analysis efficiently. Also, one of the works [21], analyzed sentiment in web videos by building a joint model that takes a combination of audio, visual and textual features.

3 DATASET AND EVALUATION

3.1 Dataset acquisition

We constructed an Indian Political Memes (IPM) Dataset for this experiment which consisted of memes that are shared on a day-to-day basis on the internet and are politically motivated. We created this dataset using the *google_images_download* [3] module which is an open-source python tool available online, to scrape several images using the keywords as the name of some famous politicians, social activists, journalists and big political events that have taken place post-independence in India. For each keyword 100 images were downloaded using the module, resulting in a corpus of images containing 5000 images. Out of this corpus of images, 1500 memes were randomly sampled and were asked to be annotated by three annotators into the following categories:
(i) Hate Inducing
(ii) Satirical
(iii) Non-offensive.

Some of the images that were blurred and consisted of no text were removed from the dataset resulting in a final dataset of 1218 memes. The memes were annotated as hate inducing if and only if the meme satisfied one or more of the following conditions: (i) meme consisted of a sexist or racial barb to malign a minority, (ii) meme had object stereotyping or (iii) meme consists of a hateful hashtag such as #HinduSc*m. The annotators were specifically asked not to consider a meme as hate-inducing due to the presence of a particular word, however offensive that word might be.

Once all the annotators had labelled each image in the dataset in one of the three categories, all the conflicts were resolved and finally, the label that was in majority was chosen as the final label for the meme. The distribution of the memes into the various classes is shown in Table 1. After final annotation, it was found that there were 427 hate-inducing memes, 452 satirical memes, 339 non-offensive memes out of the total 1218 memes dataset. Examples of non-offensive, satirical and hate-inducing memes are shown in Figure 1, Figure 2 and Figure 3 respectively. This dataset was then channelized into a pipeline that extracts the text from the images.

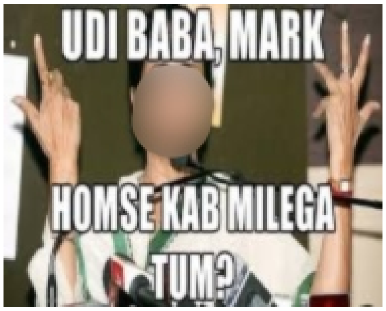

Figure 1: Non-offensive meme example from IPM dataset

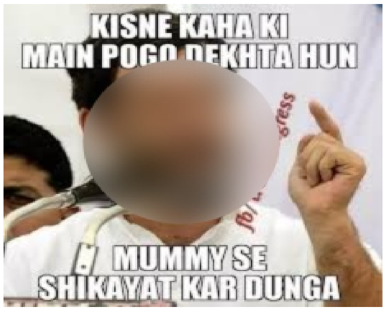
KISNE KAHA KI MAIN POGO DEKHTA HUN
MUMMY SE SHIKAYAT KAR DUNGA

Figure 2: Satirical meme example from IPM dataset

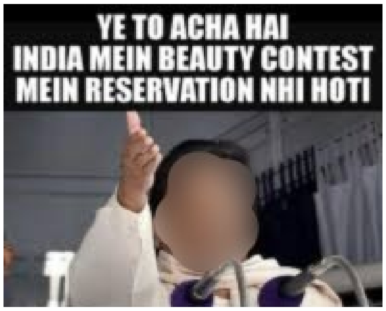
YE TO ACHA HAI INDIA MEIN BEAUTY CONTEST MEIN RESERVATION NHI HOTI

Figure 3: Hate inducing meme example from IPM dataset

3.2 Dataset Evaluation Metrics

3.2.1 Cohen's Kappa Metric. The Cohen's Kappa [7] metric is used for determining the inter-annotator agreement between two annotators. The metric determines the quality of annotation by taking into account the possibility that two annotators could have classified the subject into the same category by chance. A value of the kappa score close to 0 indicates no agreement between the

[3] http://bit.ly/2Jv55Qe

Table 2: Cohen's Kappa for three annotators A_1, A_2 and A_3

	A_1	A_2	A_3
A_1	–	0.81	0.77
A_2	0.81	–	0.87
A_3	0.77	0.87	–

annotators and a value close to 1 indicates perfect agreement between the two annotators. Mathematically, Cohen Kappa is defined by equation 1.

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \tag{1}$$

where, p_o denotes the relative observed agreement between the two annotators and p_e is the probability of chance agreement between the annotators. The Cohen's Kappa score between the three annotators for our dataset is shown in Table 2. The highest kappa score of 0.87 between annotator A_2 and annotator A_3 is shown.

3.2.2 Multilingual Index. The Multilingual Index (MI) is calculated on the text dataset which is prepared by extracting the text out of the memes of the IPM dataset. MI is used to calculate inequality in the diffusion of languages in a corpus containing two or more languages [2]. Hence it helps us to quantify the code-switching in the dataset i.e. the number of Hinglish words in the dataset. A value closer to 1 indicates good code-switching in the dataset. Let k be the total number of languages in the corpus and p_j be the fraction of the words in the language j in the corpus. Then MI is calculated mathematically by equation 2.

$$MI = \frac{1 - \sum_{j=1}^{k} {p^2}_j}{(k-1)\sum {p^2}_j} \tag{2}$$

For our dataset multilingual index was found to be 0.684 indicating good code-switching in the dataset.

3.2.3 Fleiss's Kappa Metric. The Fleiss's Kappa [12] helps to determine annotation agreement between three or more raters. Since our dataset is triply annotated, hence we calculate the Fleiss's kappa metric on our dataset. Unlike Cohen's Kappa where all annotators need to annotate all the subjects, in Fleiss's Kappa all the raters need not annotate all the subjects. Let n be the number of subjects, k be the number of categories, m be the number of annotators for each subject and x_{ij} be the number of annotators that categorize subject i to category j.

For our dataset, we calculated the Fleiss Kappa score and found it to be 0.782 indicating decent agreement between the three annotators.

4 METHODOLOGY

Our methodology primarily consists of the following steps: Preprocessing of dataset followed by training the classifier model and then using it on IPM dataset.

4.1 Preprocessing the dataset

The preprocessing consists of preprocessing of the images, extracting text from images followed by preprocessing of textual data.

4.1.1 Preprocessing Images . The image needs to be pre-processed in order to extract text out of the meme efficiently. Though the OCR performs some inbuilt pre-processing on the image, we perform the following steps for processing the image ourselves :

(i) Rescaling : Some of the images need to be rescaled to a larger size to enlarge the text written in a smaller font in the meme in order to make the text recognizable to the OCR. We use the resize feature of the OpenCV module [5] in python to rescale the images.

(ii) Gaussian blurring : The blurring effect is used to reduce noise from the image. Gaussian blurring is performed by convolving the image with a gaussian kernel. We use the Gaussian blur feature of the OpenCV module [5] to perform gaussian blurring.

(iii) Deskewing : Some of the memes have text written at some skewed angle. Deskewing helps to rotate the image such that the text written in the image is mostly horizontal.

(iv) Gaussian adaptive thresholding : Thresholding [34] converts the text into a black and white format so that it is easily recognizable to OCR. In adaptive thresholding, the gaussian mean of the surrounding area determines the threshold value for the pixel of the image.

4.1.2 Extraction of text from images . After performing the above pre-processing on the images, we pass it through an open-source OCR reader *ocr.space*[4] for extracting the text out of the memes. Table 3 depicts the samples after text extraction from the image examples given above. Also, Table 3 shows the English translation of the Hinglish text extracted from the Memes.

4.1.3 Preprocessing Text . The tweets obtained from data sources were sent through a pipeline with the objective to convert them into semantic feature vectors.

(i) Initially, the hashtags (For example: #indianpolitics), URLs, user mentions (denoted by '@') and numbers were removed from the text since they do not convey any relevant information about the sentiments of the text. Also, using the NLTK library, the stop words were eliminated.

(ii) The emoticons (For example: ":)", "XD") were replaced by their textual description about the true emotions they depict.

(iii) Many of the comments which are in *Devnagari* (Hindi) script were converted to Roman (English) script. This was done using a python library called *indic-transliterate*[5]

(iv) The Hinglish text now obtained are converted to their respective English translation using an *Xlit-Crowd Conversion Dictionary*[6].

(v) The is followed by the use of various word embedding representations such as FastText [4], Twitter word2vec [14], Glove [23] and Bert [11] embeddings for building the first layer of the LSTM side of the model which is the word-embedding layer. Different embeddings models are used to obtain the word vector representations of the preprocessed tweets. The embedding models are used one by one to figure out the best set of word embeddings.

4.2 Data Augmentation

Since the task of hate speech detection from images using deep learning requires a large number of images, the technique of data

[4] http://bit.ly/30uxnQ9
[5] http://bit.ly/2JtSc95
[6] http://bit.ly/2WVCeY8

Table 3: Example of Hinglish text extraction from the memes depicted in Figure 1, 2, 3 with their respective English translations

Figure	Hinglish Text Extracted	English Translation	Label
Figure 1	Udi baba, Mark, homse kab milega ?	Hey Mark, when will you meet us ?	Non Offensive
Figure 2	Kisne kaha ki main pogo dekhta hun. Mummy se shikayat karunga	Who said that I watch pogo. I will complain to mother	Satirical
Figure 3	Ye to acha hai India mein beauty contest mein reservation nhi hoti.	It is good that there in no reservation in beauty contests in India (Derogatory remark on personal appearance)	Hate Inducing

augmentation is used to increase the size of the dataset to train the classifier model. Data augmentation refers to methods for constructing iterative optimization or sampling algorithms via the introduction of unobserved data or latent variables [36]. We have used mainly five different types of augmentation techniques for our work which has significantly helped our model to train better on the dataset and provide much better results than what was observed previously.

The data augmentation techniques used are :

(i) Scaling : We scale the image both inwards and outwards for creating new images in the dataset.

(ii) Translation : The images are moved in the X or Y direction by varying degrees.

(iii) Rotation : Rotation is performed at 90 degrees, 180 degrees and 270 degrees.

(iv) Flipping : The images are flipped in both the horizontal and vertical directions.

(v) Adding noise : Gaussian noise is added to distort the high-frequency features that are not useful for the model.

For implementing the above techniques, we have used the various functions from the *ImageDataGenerator* class of the *Keras* image processing python library.

4.3 The Model

We propose a model which is a binary channeled CNN cum LSTM model which takes the text in the form of word vector representation and image as its input and finally concatenates the two channels to produce the final result. The model architecture is depicted in Figure 4.

4.3.1 The CNN channel. The CNN channel processes the image and tries to extract certain features of the image that would help to classify the meme into one of the three categories i.e. hate inducing, satirical and non-offensive. The pre-processed images form the input to the first layer of the CNN channel which is a convolution2D layer with filter size 64, kernel size as (5, 5) and activation function of *Relu* [22]. This is followed by a max_pooling layer of size (5, 5). The convolution layer helps to create a convolution kernel that is convolved with the input layer to produce a tensor of outputs. Next, we employ another convolution2D layer, this time of size 32, kernel size (3, 3) and activation function as *Relu*, and a max_pooling layer of pool size (3, 3). This is succeeded by a flatten layer that converts the 3-dimensional feature map to 1 dimension. We also use a dropout layer of size 0.4 to prevent overfitting of data. This is followed by a dense layer of size 32. The CNN channel tries to utilize the image features to decide in which category the meme is to be classified.

4.3.2 The LSTM channel. This particular channel is for the textual content that has been extracted from the image and preprocessed during the earlier stages of the work. The first layer of the LSTM channel is the embedding layer which takes the word vector representation of the extracted caption from the Meme image. These embeddings help to learn distributed representations of captions. After experimentation, we kept the size of embeddings fixed to 100. Different embedding models unravel the different aspects of the language. For example, the dependency parser focuses on the similarity between the two terms. On the other hand, statistics of bag of word (BoW) embeddings emphasize on word associations. Some of the embeddings used are FastText [4], Glove [23], Twitter word2vec [14] and Bert [11]. The embedding layer is followed by a dropout layer of size 0.2 to prevent overfitting of data. The next is the LSTM layer of size 64 with a dropout of 0.4. The LSTM layer is followed by two dense layers of sizes 64 and 32. This part of the model serves as a processing model for the textual content.

4.3.3 Recombination of channels. The two parallel channels of the model which process the text and images separately are finally recombined to a single channel to obtain the final results. The concatenation is followed by the presence of two dense layers of sizes 32 and 3. The last dense layer of size 3 uses *softmax* as the activation function. We use L2 regularization and *Adam* optimizer [17] for preventing overfitting. The loss function used in the last layer was categorical cross-entropy which serves beneficial in the case of multiple classes. The output obtained after passing through the dense layers is one of the three classes, i.e., Hate Inducing, Satirical and Non-offensive. The two parts of the model, therefore, help to process the text and image in parallel thereby tackling the task of hate speech detection in a much formal and organized manner.

5 EXPERIMENTATION AND RESULTS

In this section, we analyze the results of various models on the IPM dataset. As a baseline, we first conduct experiments using supervised machine learning models, namely SVM and random forest classifier which will define the baseline results. We then use LSTM and CNN models followed by the analysis of our proposed model.

5.1 Baseline

The baseline model was created using a Support Vector Machine (SVM) and a Random Forest (RF) classifier. These two classifier models were trained using k-fold cross-validation with 10 splits. For the SVM classifier, we choose kernel value as ‘poly’ with the default

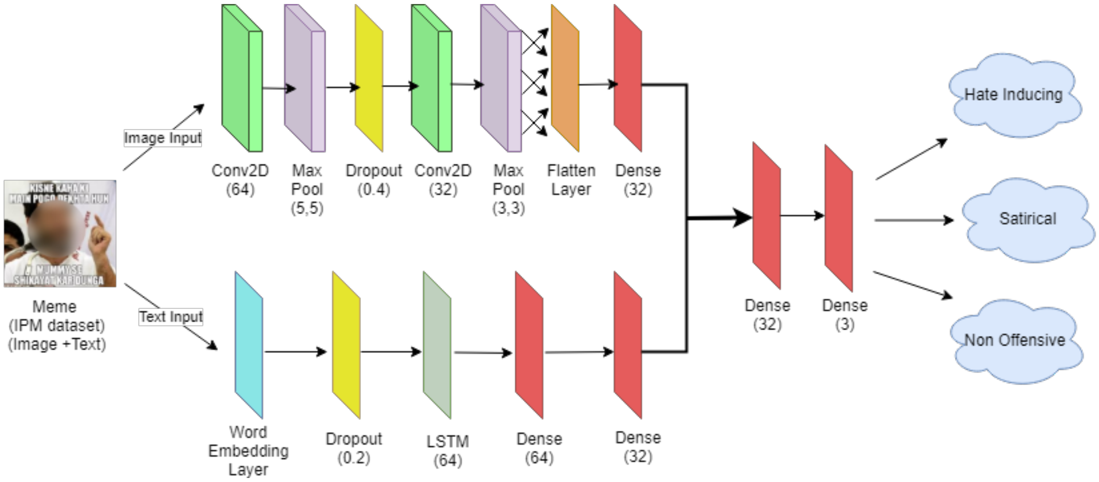

Figure 4: Proposed model architecture

Table 4: Baseline results for non-offensive, hate-inducing, satirical memes classification on IPM dataset using SVM and random forest classifier with different features

Feature	GLCM		Colorfulness		Tamura		Human Face	
Classifier	SVM	RF	SVM	RF	SVM	RF	SVM	RF
Precision	**0.622**	0.584	0.542	0.475	0.562	0.512	0.608	0.545
Recall	**0.651**	0.597	0.522	0.457	0.592	0.538	0.619	0.595
F1-Score	0.634	0.575	0.573	0.514	0.583	0.546	**0.658**	0.603

Table 5: Results of Deep learning models on IPM dataset

Result	**CNN Model**	**LSTM Model**
Precision	0.632	0.581
Recall	**0.674**	0.534
F1 - Score	0.618	0.604

value of degree = 3 as hyperparameters. All other hyperparameters for the SVM classifier are used at their default values. For the RF classifier after fine-tuning the model, the hyperparameters chosen were n_estimators as 600, max_depth as 12 and max_features as *log2*. We choose the following features from the images to be input in the baseline classifying models:

(i) GLCM features : Gray-Level Co-occurrence Matrix helps to determine the texture of the image which is useful for determining the emotional expression in an image. We use GLCM for determining the contrast, correlation, energy, and homogeneity of an image [15] which will be used as features for our model.

(ii) Colorfulness feature: Color is one of the important ways to convey a message through an image. Colorfulness is calculated using Earth Mover's Distance (EMD) between the histogram of an image and the histogram having a uniform color distribution [8].

(iii) Tamura features : Tamura features also help for determining the texture of an image as shown by [35] . We use coarseness and directionality as Tamura features for input to the classifying model.

(iv) Human face feature : Human faces are important for drawing attention to a meme. The number of human faces and the size of human faces are used as features of our classifying model. [37].

We use these features as input to the classifying SVM and RF model with hyperparameters mentioned above. The results using these baseline models and the features mentioned are shown in Table 4. We use precision, recall and F1-score as the metrics for determining the baseline results. It was seen that SVM performs marginally better than the Random forest classifier when using GLCM features. The SVM classifier also gave comparable results when using the human face features as the input. This forms our baseline results for Hinglish offensive memes classification on IPM Dataset.

5.2 Deep learning models

We also compare our model to some of the deep learning models. We use two deep learning models for validating the accuracy of our own model. The first model we experimented with, is a CNN-based model which is popular for image classification. The CNN model was trained using a k-fold cross validation with 10 splits. The first layer of the CNN model is a Convolution2D layer with filter size 64 and kernel size = (5, 5). The activation function used is *Relu* [22] and a max_pooling layer of pool size = (5, 5) is employed. This is followed by another Convolution2D layer with a filter size of 32 and kernel size of (3, 3), succeeded by a max-pooling layer of pool size = (3, 3). We also employ a dense layer of size 64 and a dropout layer of 0.4 to prevent overfitting. The hyperparameters are chosen with the help of grid search which helps to select those hyperparameters that produce the most optimal results.

The second model experimented with is an LSTM based model which consists of an embedding layer, followed by an LSTM layer and two dense layers of size 64 and 32. Adam optimizer and L2 regularization are used to prevent overfitting of data. The dropout layer of size 0.4 is also added to prevent overfitting of data. The LSTM based model is also trained using k-fold cross-validation with 10 splits. The loss function used for both the CNN-based model and the LSTM based model is categorical cross-entropy.

Table 6: Results of the our model with different flavors of word embeddings.

Features	Precision	Recall	F1
Glove (Gl)	**0.762**	**0.816**	**0.792**
Twitter Word2vec (Tw)	0.741	0.766	0.781
FastText (Ft)	0.721	0.694	0.773
Bert (Bt)	0.758	0.804	0.784
(Gl) + (Tw)	0.727	0.751	0.722
(Gl) + (Ft)	**0.798**	**0.779**	**0.764**
(Tw) + (Bt)	0.748	0.740	0.702
(Gl) + (Bt)	0.779	0.790	0.725
(Bt) + (Ft)	0.760	0.723	0.746

We compare the results of the two deep learning models in Table 5 using precision, recall and F1-score as the metrics. The CNN-based model produces much better results than the LSTM based model, while the results are marginally better than the SVM baseline model described above. This is due to the fact that hate speech can be conveyed in the form of text as well as images. Analyzing the image solely therefore does not produce great results for this classification task. Hence, we see that the features extracted manually and fed to SVM or RF classifier give comparable results to that of CNN or LSTM based models. As proposed in our model, we analyze both images as well as the text extracted to produce the final results.

5.3 Our Model

Now we compare the results of the baseline (SVM and RF) model and the deep learning models (CNN and LSTM) to our proposed model. Our model tries to incorporate the best features of both the CNN and LSTM deep learning models by considering images as well as the extracted text from the images for the classification task. We propose a binary channel model in which the LSTM channel processes the text written inside the images and the CNN channel processes the image itself. The two channels are combined to produce the final results. We conduct the experiments on our model using different flavors of word embeddings i.e. (i) Twitter Word2vec [14] (ii) Glove [23] (iii) Fastext [4] and (iv) Bert [11] embeddings as well as the combination of the above embeddings. Many different sizes of embeddings were tested. Finally, the size of the embeddings was chosen to be 100. Our model is also trained using k-fold cross-validation with 10 splits to maintain consistency in all the experiments.

The results of our model using different types of word embeddings on the IPM dataset are shown in Table 6. The results are compiled using precision, recall and F1-score as metrics of evaluation. Our model outperforms the baseline (SVM and RF models) and also the deep learning (CNN and LSTM) models, hence establishing itself as the state of the art for the task of Offensive memes classification in Hinglish language. As seen from Table 6, the best results obtained using a single embedding model were with the Glove embeddings. A recall score of 0.816 was recorded with Glove embeddings. Also, experiments were conducted using the combination of word embeddings where (Glove + Fastext) is seen to produce the best results. Here, a precision of 0.798 was obtained which demonstrates that the model outperforms all the other models on the IPM dataset for the task of offensive memes classification in code switched language (Hinglish).

5.4 Error Analysis

We analyze the possible reasons due to which our model gives error in its judgment.

(i) OCR error : We have used ocr.space for extracting the text out of the memes. The memes in which the text is written in very small font, or the text written is slightly blurred, the OCR fails to recognize that text with 100% accuracy. Also, in many cases, it is hard for the OCR reader to extract text which is written vertically or diagonally.

(ii) Unconventional words (code switched) : A little work is done in dealing with uncommon Hinglish words which may arise due to spelling variations, grammatical errors or mixing of some regional languages by the creators of the memes. For example, the spelling variation resulting from a difference in the pronunciation of the words can create a new set of words that are not present in the dictionary itself.

(iii) Disguised hate : Some memes are designed so that they might seem to be satirical to the annotators but might actually be inducing hate towards an individual in a disguised fashion. Such memes would not be correctly classified by our model. For example, *"Abbe oh, ma***sa jane vale"* which translates to *"Hey, religious school going person"*

(iv) Overfitting of data : Due to the training using deep learning models and also the memes on social media being repetitive, there might be some overfitting of data. We have tried to avoid the problem of overfitting by using the dropout layers and the best set of hyperparameters. However, the problem might still be present and may cause variation in the results.

6 CONCLUSION

In this paper, we introduced a novel dataset, i.e the IPM (Indian Political Memes) Dataset which consists of images (Memes) classified in three categories - Benign, Satirical and Hate Inducing. Also, we proposed a pipeline to detect offense and hate from images that contain text in code-switched languages. We developed a multichannel CNN-LSTM model, which processes the images and text individually and combines the analysis from both channels to give the final classification result. The model plucks out the text from images and converts the text into a word vector representation before passing through the LSTM channel of the model. A number of different words embedding models are tried to attain the most optimum results. On the other hand, the images are passed through the CNN channel of the model. We compare the results of our model to other deep learning-based models and some supervised machine learning models, namely SVM and Random Forest classifier after extracting features from the images. The results suggest that our model outperforms all the other models producing state-of-the-art results for Hinglish Language Memes classification on the IPM dataset. These results also mark the fact that parallel analysis of text and image gives much better outcomes as compared to processing of images alone. We also release the code, the dataset made and the model proposed in our work. We believe this method would be useful for hate speech detection for images in code-switched languages.

7 FUTURE WORK

This work can be further extended to various avenues like the inclusion of videos as a part of our analysis. A political speech video can be categorized as offensive by analyzing the speech and the video graphics inspired by the work of [27]. The existing methods can also be used by various social media platforms to classify the complete page or user as objectionable. The pages on Facebook or Twitter users who pop up during election times to provoke the masses can be detected and removed. Other GRU-based models [43] can also be applied to this problem. Additionally, we can use our model for other code-switched language pairs. For example, the work by [38] detects sentiments from Chinese-English pair of code switched language. Also, the relative positions of words play a major role in the analysis of the Hinglish text. So, we would like to explore such possibilities in our future work [30]. We can also consider building a hate-inducing video segmentation system [28, 29] in order to remove hate-inducing videos from the internet.

REFERENCES

[1] Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. " I am borrowing ya mixing?" An Analysis of English-Hindi Code Mixing in Facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching.* 116–126.

[2] Ruthanna Barnett, Eva Codó, Eva Eppler, Montse Forcadell, Penelope Gardner-Chloros, Roeland Van Hout, Melissa Moyer, Maria Carme Torras, Maria Teresa Turell, Mark Sebba, et al. 2000. The LIDES Coding Manual: A Document for Preparing and Analyzing Language Interaction Data Version 1.1–July 1999. *International Journal of Bilingualism* 4, 2 (2000), 131–271.

[3] Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection. In *Proceedings of the Second Workshop on Computational Modeling of PeopleâĂŹs Opinions, Personality, and Emotions in Social Media.* 36–41.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

[5] Gary Bradski and Adrian Kaehler. 2008. *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.".

[6] Guoyong Cai and Binbin Xia. 2015. Convolutional neural networks for multimedia sentiment analysis. In *Natural Language Processing and Chinese Computing.* Springer, 159–167.

[7] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[8] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. 2006. Studying aesthetics in photographic images using a computational approach. In *European conference on computer vision.* Springer, 288–301.

[9] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh International AAAI Conference on Web and Social Media.*

[10] Fabio Del Vigna12, Andrea Cimino23, Felice DellâĂŽOrletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on Facebook. (2017).

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[12] Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76, 5 (1971), 378.

[13] Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online.* 85–90.

[14] Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia Lab @ ACL WNUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations. In *Proceedings of the Workshop on Noisy User-generated Text.* 146–153.

[15] Robert M Haralick and Linda G Shapiro. 1992. *Computer and robot vision.* Vol. 1. Addison-wesley Reading.

[16] Raghav Kapoor, Yaman Kumar, Kshitij Rajput, Rajiv Ratn Shah, Ponnurangam Kumaraguru, and Roger Zimmermann. 2018. Mind Your Language: Abuse and Offense Detection for Code-Switched Languages. *arXiv preprint arXiv:1809.08652* (2018).

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.* 1097–1105.

[19] Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. 2018. Did you offend me? classification of offensive tweets in hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2).* 138–148.

[20] Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. 2018. Detecting offensive tweets in hindi-english code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media.* 18–26.

[21] Louis-Philippe Morency, Rada Mihalcea, and Payal Doshi. 2011. Towards multimodal sentiment analysis: Harvesting opinions from the web. In *Proceedings of the 13th international conference on multimodal interfaces.* ACM, 169–176.

[22] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10).* 807–814.

[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).* 1532–1543.

[24] Matthew Pittman and Brandon Reich. 2016. Social media and loneliness: Why an Instagram picture may be worth more than a thousand Twitter words. *Computers in Human Behavior* 62 (2016), 155–167.

[25] Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Amir Hussain. 2016. Convolutional MKL based multimodal emotion recognition and sentiment analysis. In *2016 IEEE 16th international conference on data mining (ICDM).* IEEE, 439–448.

[26] Kshitij Rajput, Raghav Kapoor, Puneet Mathur, Hitkul, Ponnurangam Kumaraguru, and Rajiv Ratn Shah. 2020. *Transfer Learning for Detecting Hateful Sentiments in Code Switched Language.* Springer Singapore, Singapore, 159–192. https://doi.org/10.1007/978-981-15-1216-2_7

[27] Verónica Pérez Rosas, Rada Mihalcea, and Louis-Philippe Morency. 2013. Multimodal sentiment analysis of spanish online videos. *IEEE Intelligent Systems* 28, 3 (2013), 38–45.

[28] Rajiv Ratn Shah, Yi Yu, Anwar Dilawar Shaikh, Suhua Tang, and Roger Zimmermann. 2014. ATLAS: automatic temporal segmentation and annotation of lecture videos based on modelling transition time. In *Proceedings of the 22nd ACM international conference on Multimedia.* ACM, 209–212.

[29] Rajiv Ratn Shah, Yi Yu, Anwar Dilawar Shaikh, and Roger Zimmermann. 2015. TRACE: linguistic-based approach for automatic lecture video segmentation leveraging Wikipedia texts. In *2015 IEEE International Symposium on Multimedia (ISM).* IEEE, 217–220.

[30] Anwar D Shaikh, Mukul Jain, Mukul Rawat, Rajiv Ratn Shah, and Manoj Kumar. 2013. Improving accuracy of sms based faq retrieval system. In *Multilingual Information Access in South Asian Languages.* Springer, 142–156.

[31] Stefan Siersdorfer, Enrico Minack, Fan Deng, and Jonathon Hare. 2010. Analyzing and predicting sentiment of images on the social web. In *Proceedings of the 18th ACM international conference on Multimedia.* ACM, 715–718.

[32] Sara Owsley Sood, Elizabeth F Churchill, and Judd Antin. 2012. Automatic identification of personal insults on social news sites. *Journal of the American Society for Information Science and Technology* 63, 2 (2012), 270–285.

[33] Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In *Aaai/iaai.* 1058–1065.

[34] Chris Stauffer and W Eric L Grimson. 1999. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, Vol. 2. IEEE, 246–252.

[35] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. 1978. Textural features corresponding to visual perception. *IEEE Transactions on Systems, man, and cybernetics* 8, 6 (1978), 460–473.

[36] David A Van Dyk and Xiao-Li Meng. 2001. The art of data augmentation. *Journal of Computational and Graphical Statistics* 10, 1 (2001), 1–50.

[37] Paul Viola and Michael J Jones. 2004. Robust real-time face detection. *International journal of computer vision* 57, 2 (2004), 137–154.

[38] Zhongqing Wang, Sophia Lee, Shoushan Li, and Guodong Zhou. 2015. Emotion detection in code-switching texts via bilingual and sentimental information. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Vol. 2. 763–768.

[39] Wang Wei-ning, Yu Ying-lin, and Jiang Sheng-ming. 2006. Image retrieval by emotional semantics: A study of emotional space and feature extraction. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4. IEEE, 3534–3539.

[40] Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. 2012. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In

Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 1980–1984.

[41] Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian D Davison, April Kontostathis, and Lynne Edwards. 2009. Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB* 2 (2009), 1–7.

[42] Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang. 2015. Robust image sentiment analysis using progressively trained and domain transferred deep networks. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[43] Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European Semantic Web Conference*. Springer, 745–760.