Empowering First Responders through Automated Multimodal Content Moderation

Divam Gupta
IIIT-Delhi
divam14038@iiitd.ac.in

Indira Sen
IIIT-Delhi
indira15021@iiitd.ac.in

Niharika Sachdeva
IIIT-Delhi
niharikas@iiitd.ac.in

Ponnurangam Kumaraguru
IIIT-Delhi
pk@iiitd.ac.in

Arun Balaji Buduru
IIIT-Delhi
arunb@iiitd.ac.in

***Abstract*— Social media enables users to spread information and opinions, including in times of crisis events such as riots, protests or uprisings. Sensitive event-related content can lead to repercussions in the real world. Therefore it is crucial for first responders, such as law enforcement agencies, to have ready access, and the ability to monitor the propagation of such content. Obstacles to easy access include a lack of automatic moderation tools targeted for first responders. Efforts are further complicated by the multimodal nature of content which may have either textual and pictorial aspects. In this work, as a means of providing intelligence to first responders, we investigate automatic moderation of sensitive event-related content across the two modalities by exploiting recent advances in Deep Neural Networks (DNN). We use a combination of image classification with Convolutional Neural Networks (CNN) and text classification with Recurrent Neural Networks (RNN). Our multilevel content classifier is obtained by fusing the image classifier and the text classifier. We utilize feature engineering for preprocessing but bypass it during classification due to our use of DNNs while achieving coverage by leveraging community guidelines. Our approach maintains a low false positive rate and high precision by learning from a weakly labeled dataset and then, by learning from an expert annotated dataset. We evaluate our system both quantitatively and qualitatively to gain a deeper understanding of its functioning. Finally, we benchmark our technique with current approaches to combating sensitive content and find that our system outperforms by 16% in accuracy.**

***Index Terms*—Multimodal detection, Natural language processing, Image analysis**

I. Introduction

Objectionable content is rampant on Online Social Networks (OSNs) and can cause significant harm especially during crisis events. Furthermore, content of this nature often transcends virtual borders and causes repercussions 'offline', affecting the lives of the post creator and those around her [15]. When the Assam riots broke out in 2012 in India, messages circulated over social media caused widespread panic and an exodus of Indian people of North Eastern origin from other Indian cities.[1] To mitigate the harmful effect of objectionable content during crises, it is essential that first responders[2] have access to such content.

While such objectionable content is often illegal or contraband according to the social media platform's guidelines, they remain on OSNs since it is hard to detect such content. Organizations often hire special third-party services to flag such content [1]. There are drawbacks to this model of community moderation since the moderator lacks an all-encompassing code of what exactly constitutes as sensitive content. For example, while according to Twitter's guidelines [3] stalking is not listed as unlawful, it is mentioned in Youtube's rulebook [4] as an offence. Even after honing on the exact nature of the content that is to be flagged, a moderator's search space is expansive due to the sheer volume of posts generated. Manual inspection is therefore not scalable. This is especially true if such content needs to be detected and nullified in a time-sensitive manner. Finally, prolonged exposure to content of this nature can cause long-lasting psychological effects [2].

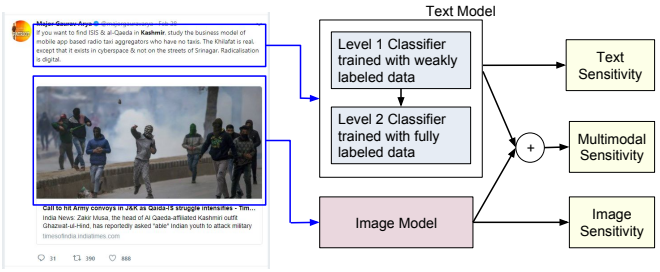

Fig. 1. Highlighted in the image is a tweet which has both visual and textual components. Each of these are fed to a respective classifier whose output is then fused to obtain a combined sensitivity verdict.

It is, therefore, vital to have an automated detection system which reduces the search space and cognitive load of the moderator. While previous research has attempted to fill this gap through detection of hate-speech and abusive content [18], much of the work focuses on dealing with textual content and some image content [12]. However, a post might contain both components simultaneously as we see in the tweet in Figure 1. Furthermore, previous research has shown that in many tweets, the image contains useful information which is not present in the text [30]. Therefore current automatic detection systems fail to deal with such multimodal content. There are also tools aimed towards enabling first responders to gain access to situational information during crises, but these tools do not focus on sensitive or objectionable content [21]. To that

[1]https://in.reuters.com/article/bangalore-assam-north-east-bodo/thousands-flee-bangalore-over-assam-violence-idINDEE87F0BU20120816

[2]First responders include law enforcement, aid workers and other officials providing service during emergencies.

end, we believe our work is *the first attempt at automatically detecting multimodal sensitive content for first-responders as summarized in Figure 1*.

To achieve this goal to create an automatic content moderation system we first curate a set of guidelines for sensitive content detailed in Section 2. We introduce state-of-the-art baselines in Section 3 as well as recent advances in multimodal classification and abusive content detection. In Section 4, we describe our two-level text classification framework as well as the image classification model, before presenting how these models are fused together to form a multimodal system which classifies tweets. Finally, we present a detailed evaluation of our system, comparisons with existing baselines and a qualitative analysis of the results in Section 5. Our model achieves an accuracy of 80.13% and 16.63% increase in AUC compared to a state-of-the-art baseline.

II. Background

What is sensitive content for first responders? Sensitive content is *in violation of compliance with legal and regulatory exigencies, site/community guidelines, user agreements, and that it (fails to) fall within norms of taste and acceptability for that site and its cultural context*.[3] A key challenge in detecting sensitive content is the subjectivity of its definition. To reduce ambiguity, we enumerate items that various websites believe to be objectionable. We then restrict ourselves to studying only one aspect of such content: of interest to first responders.

We begin by consulting first responders in India about what kind of content they would like to be automatically detected. Based on their inputs, we create a set of guidelines. To do so, we start by collecting the community guidelines or set of rules governing conduct on 500 most reputed web platforms as listed by Alexa.[4] Each differently worded set of rules is assigned a broad category using topic modeling (LDA). A few of the categories and the corresponding entry for an OSN is shown in Table I. These broad categories form the main guideline for annotations of social media content by experts. Since our work is tailored to aiding first responders, we focus on a subset of sensitive categories that is oriented towards them. Two experts review our rulebook to resolve any differences through discussion and create the final rulebook as shown in Table II.

Finally, we validate our rulebook's completeness concerning first responders in India by interviewing 22 Law Enforcement officials at various Government and Non-government agencies including the ones who contributed in the initial interviews. Note that such a process is context dependent but the general framework can be reused in a different setting with a small amount of input from domain experts (first responders in our case) of that setting.

III. Related Work

In this section we present work closely related to ours on two axes - a) Detecting sensitive, controversial or toxic content on Social Media to draw from common themes present in these works and b) classification frameworks for textual, visual and multimodal content.

TABLE I
Items considered sensitive or objectionable by various OSNs. We focus on the bolded topics since they directly impact first-responders.

Topic	Source Website(s)
Violence	Twitter, Facebook, Google+, Youtube, Amazon
Sexually Explicit Content	Twitter, Facebook, Google+, Youtube, Amazon
Medical Procedures / Health Disclosures	Twitter, Yahoo
Hate Speech	Facebook, Youtube, Yahoo, Amazon, Indian Penal Code (IPC)
Political messages	Youtube, Yahoo, IPC
Personally Identifying Information	Google+, Amazon
Terrorism / Mobilisation	Youtube, Amazon, IPC
Harassment	Facebook, Yahoo, Amazon

Detecting Objectionable Content. Warner et al. provided one of the first comprehensive studies on abusive content propagation, particularly anti-semitic hate speech [27]. Since then multiple studies have contributed to this space. Closest to our work is Nobata et al's study of a large annotated corpus of Yahoo comments, used in conjunction with syntactic features to detect abusive content [18] . While our work explores similar themes, our two-step labeling process, one level which uses hashtags for weak labeling and the second level harnessing expert domain knowledge on a smaller dataset is less resource hungry while employing a weakly supervised approach towards sensitive image classification which has not been studied previously. Finally, toxic posts, a subset of sensitive content, are analyzed by Wulczyn at al. in [29] using large-scale human annotations. The researchers use crowdflower to annotate comments on Wikipedia and use a Logistic Regression model trained on this labeled data to automatically classify more comments. We use this study (referred to as ToxicBaseline from now on) as a baseline evaluation in section 5. Various forms of deviant behavior have been widely studied such as cyberbullying [11], harassment [8] but there seems to be a lack of solutions-oriented for first responders. Rudra et al. [21] aims to create effective tools for First Responders during crises to access situational information that might improve aid efforts, while Brengarth et al. enables cooperation between aid workers and citizens plagued by disaster in [6].

Content Classification Approaches and Multimodal Approaches. Our work builds on Natural Language Processing (NLP) based approaches for content analysis such as detecting fake news [24], rumours [17] and disinformation [14]. The given techniques use the features of the text such as Parts-of-Speech (POS) structure and presence of named entities (NEs) for classification. Our use of deep models ensure that we do not have to rely on feature engineering. Our work is therefore thematically similar to the use of Recurrent Neural

[3]https://illusionofvolition.com/behind-the-screen/
[4]https://www.alexa.com/siteinfo

TABLE II
RULEBOOK OF SENSITIVE CONTENT CATEGORIES AND WHAT THEY ENTAIL

Given a post, mark if it is sensitive or not, if yes then mark the subcategory it belongs to	
Hate speech	shows disrespect on grounds of religion, race, place of birth, residence, language,caste or community to a person or a group - if you voted for these people, YOU are responsible for this absurd nonsense! No #AchcheDin, just #NoKachecheDin.
Violence or Gore	violent or gory content that is shocking, sensational, or disrespectful. - Shoot Anti National JNU students & Proff. And close the chapter But one Problem is that Our Govt. doesn't allow to kill Dogs #WakeUpIndia
Political Criticism	brings or attempts to bring into hatred or contempt, or excites or attempts to excite disaffection towards the Government - We request our Pakistani Brothers to trend #SupportJNU for our pro-Pakistani JNUites borthers #PakStandWithJNU
Mobilisation	seeks to organize a movement or protest - Anti-India protest in #Pularma Kashmir today, come one come all #KashmirKillings #KashmirinSiege

Network (RNN) for detecting rumours [16]. We extend the use of RNNs to detect sensitive content and combine it with an image model. Convolutional Neural Networks (CNN) [13] are widely used for image classification. Previous work has used CNNs to detect hoax images [5]. Chancellor et al [7] look at the multimodal classification of pro-eating disorder content using a fusion model but relies completely on expert generated ground truth for both textual and visual posts. In contrast, our method bootstraps a small amount of expertly annotated data (for text) and weakly labeled samples (image and text) to train a model.

IV. CLASSIFICATION FRAMEWORK

In this section, we provide a detailed description of multiple components of our multimodal content classification framework. Our overall framework contains a two-step classification module, where the first is trained to detect sensitive topics, while the second classifier eliminates false positives. The multimodal classifier is a fusion of the text classifier and the image classifier trained jointly. Figure 2 summarizes the architecture of our system.

A. Data Collection

We collect two different datasets for the two-level classifiers. The data for the level 1 classifier is of greater quantity and is weakly labeled using the hashtags. The data for the level 2 classifier is labeled by human annotators in the process. We also collect a dataset of sensitive images for training the image sensitivity model.

Collecting Tweets for level 1 classifier. For training the level 1 classifier, we need a substantial amount of data to filter the tweets which have a high chance of being sensitive. We collect tweets from various hashtags and annotate them at hashtag level. We use trendogate.com [5] to collect various hashtags popular in India. For all the hashtags collected, we select the hashtags for which data is strongly inclined to being either sensitive or non-sensitive, so we use only those hashtags out of which a sample majority of randomly sampled 50 tweets fall in at least one category. We have 82 such hashtags, and 361,151 sensitive and 1,344,560 nonsensitive tweets are collected. We observe that most of the sensitive tweets are from hashtags of events related to protests, uprising or violoent movements. The list of a few filtered hashtags is mentioned in Table III.

[5]https://trendogate.com

TABLE III
TOP HASHTAGS FOR OUR LEVEL 1 CLASSIFIER DATASET

Sensitive Hashtag	# Tweets	Non Sensitive Hashtag	# Tweets
AsaramBapuji	190696	Nifty	202894
Freekashmir	74237	IndvsSA	136096
3rdhinduadhiveshan	38823	MondayMotivation	110178
Owaisi	33098	IPLfinal	103083
lovejihad	24297	MWC16	92309

Collecting Tweets for level 2 classifier. For the level 2 classifier, data with precise annotations is required. To that end, we collect tweets from sensitive events and get them annotated by human experts. We collect 7,886 tweets using the hashtags of various sensitive events. Two annotators manually label these tweets based on the rules listed in table II. After calculating the inter-annotator agreement using Krippendorf's α , since it is suitable for likert scale based tasks [10], (0.73), 6,198 tweets were labeled sensitive, and 1,688 as nonsensitive. The list of the hashtags for the level 2 classifier is mentioned in Table IV.

Data for the image classifier. We use a combination of methods to collect a dataset of 4,500 sensitive and nonsensitive images. We first download images using search engines such as Google Image search. We curate several search queries for an event as the number of results per search query accessible to us are limited. Like the hashtags in the previous section, queries are crafted keeping in mind the sensitive events, therefore, we select protest and militancy-related events for data collection.

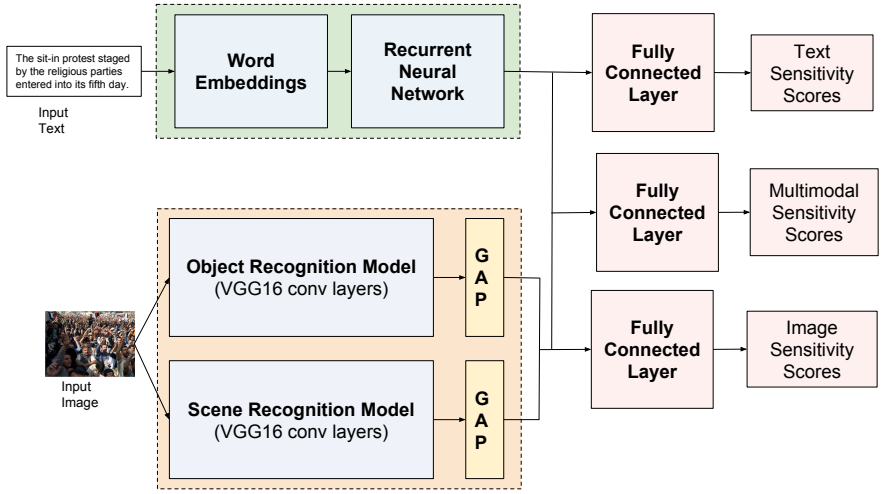

Fig. 2. The multimodal architecture is the fusion of the text model and the image model. The fused features are passed to another FC layer to get the sensitivity scores. The first, second and third FC layers are for computing the text sensitivity, multimodal sensitivity and image sensitivity scores respectively.

TABLE IV
OVERVIEW OF THE DATASET COLLECTED AND ANNOTATED FOR THE LEVEL 2 CLASSIFIER.

Hashtag	Sensitive Tweets	Non Sensitive Tweets
CauveryProtest	2129	796
JaichandKejriwal	768	270
DhakaEid	1280	64
TamilNaduBandh	334	85
Kashmir	358	110
Jallikattu	1329	363

On manual analysis, we find that approximately 50% of the images collected are were not related to the event. To remove the unrelated images, we first extract the tags of each image using the Clarifai API. [6] After manual qualitative inspection, we observed that Clarifai generated better tags compared to other API's such as Google's Cloud Vision API . We then fit a bag-of-words Naive Bayes model to classify if the given set of tags for an image are of the given event or not. Each image sample is represented as a bag of words vector from the extracted tags. For training, positive class samples are the images downloaded for that event, and negative samples are random images downloaded from various news websites. We then take the top K tags whose weights assigned by the classifier are maximum and discard all the images that don't contain those given tags. Following this process, we are left with images that are of that given event with a high probability. The Clarifai API is only used to clean the training dataset and not used in the system during the runtime. After cleaning the data, we observe that 95% of the images in the dataset are correctly labeled. Figure 3 shows the block diagram of the image collection pipeline.

[6]https://www.clarifai.com/api

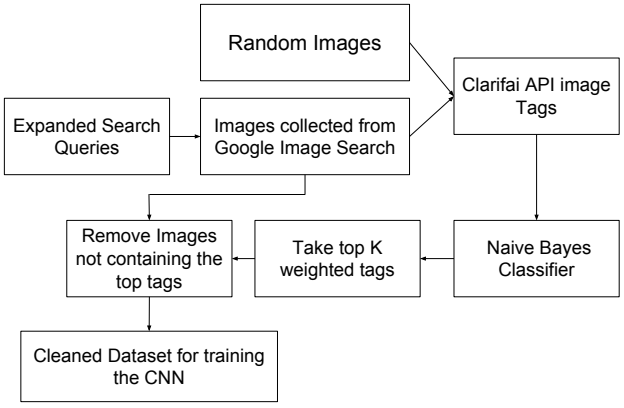

Fig. 3. The pipeline for collecting sensitive images. Google image search is used to collect the images and Clarifai API is used to remove the outliers.

B. Network Architecture

Our multimodal classifier takes both the visual and textual content of a tweet into account in classifying the tweet. The classifier is composed of an image model and a text model. The image model is based on a Deep Convolutional Neural Network (CNN), and the text model is a based on a Recurrent Neural Network (RNN). The fusion model is trained jointly on the text and the image data.

Text Model. RNNs have shown good performance for various natural language processing tasks such as sentiment analysis, [25], question answering and machine translation. The main advantage of the recurrent models over the Bag-of-Words (BoW) models is that they capture the relationships between the words of the sentence. We explore LSTM, GRU and Multi-layered GRU. Compared to LSTM, GRU contains a lesser number of parameters, which prevents over-fitting.

Preprocessing. We anonymize all the entities in the text of the tweet which would be specific to a particular event or a hashtag. The anonymization prevents the model from getting biased towards specific names, places or hashtags. We use Named Entity Recognition [20] to replace all the names, places and dates with a special tokens `NAME`, `PLACE` and `DATE` respectively. We also remove all the hashtags, mentions and URLs from the tweet.

Each word of the processed text is vectorized to word embeddings. In particular, given a sentence of N words, $\mathbf{S} = \{w_1, w_2, ..., w_N\}$ our model predicts the probability of sensitivity label $y_{sensitive}$. The operations of the GRU model can be summarized as follows. For each time step t, h_t and e_t are the output and the input vector respectively. Given e_t and h_{t-1}, h_t at time step t the updates of the GRU are:

$$z_t = sigm(W_z u_t + U_z h_{t-1} + b_z) \quad (1)$$

$$r_t = sigm(W_r u_t + U_r h_{t-1} + b_r) \quad (2)$$

$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot tanh(W_h e_t + U_h(r_t \cdot h_{t-1}) + b_h) \quad (3)$$

where z_t and r_t are the update gate vector and the reset gate vector respectively. The probability of sensitivity label $y_{sensitive}$ is computed as $P(y_{sensitive}) = sigm(W_C \times h_T)$ where $W_C \in \mathbb{R}^{1 \times d^{\text{hidden}}}$, d^{hidden} is the dimension of the output vector from GRU and T is the last time step.

We create a fixed size word vocabulary V of most frequent words. Each word w_i is represented as one-hot vector v_i of size $|V|$ where we have 1 at the position of the word and zeros at all the other places. All the words not in the vocabulary are replaced by a special token `UNK`. All the words of the sentence are mapped to embedding vectors $\{e_1, e_2, ..., e_N\}$ where each $e_i \in \mathbb{R}^{d^{emb}}$. We use an embedding matrix $W_{emb} \in \mathbb{R}^{d^{emb} \times |V|}$ and the embeddings are computed as $e_i = W_{emb} \times v_i$. The embedding matrix W_{emb} is initialized with random values and updated via backpropagation while training the complete model on the final objective.

The embedding vectors of each word are fed into the RNN one word after the another. We take the output vector of the RNN at the last step T and pass it to a fully connected layer with sigmoid activation. In particular, we use Gated Recurrent Unit (GRU) Neural Networks for modeling the sequences. Using the output $h_T \in \mathbb{R}^{d^{\text{hidden}}}$ from the GRU at the last step, the probability of sensitivity $y_{sensitive}$ is computed as $P(y_{sensitive}) = sigm(W_C \times h_T)$ where $W_C \in \mathbb{R}^{1 \times d^{\text{hidden}}}$. In the implementation we take $N = 30$, $d^{emb} = 150$ and $d^{\text{hidden}} = 512$.

Image Model. Convolutional Neural Networks (CNN) have shown competitive performance in many computer vision tasks such as image classification, object detection, image segmentation. We use a multi-branch CNN model for the task of classifying sensitive images.

From previous literature, important features for detecting sensitive images are the objects present and the type of place where the event is occurring. The features from the scene recognition model contain the information for the related to the surroundings, type of location, time of the day, etc. The features from the object recognition model contain information of all the objects present in the image. We extend the work of Wang et al. which proposes an event recognition model, using a scene and an object model [26]. For the object recognition model and scene recognition model we use VGG16 pre-trained on LSVRC challenge dataset [22] and the MIT Places 2 [32] dataset respectively. For both the VGG16 models, we only use the Convolution Layers with the weights initialized by their respective pre-trained models. If we reuse the fully connected layers of VGG16, we are restricted by the default image size of VGG16 which is 244 x 244 providing less flexibility when it comes to images of other dimensions. In the VGG16 model about 80% of the parameters are in the FC layer, hence by removing those FC layers, we are reducing the number of parameters and preventing overfitting. Figure 4 shows the architecture diagram of the image classification model.

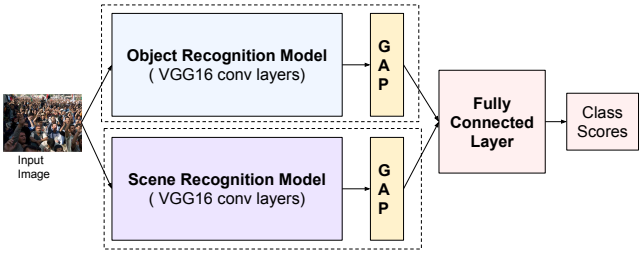

Fig. 4. The Image Sensitivity model combines the from the object recognition and the scene recognition model and predicts the class scores.

In particular given an image I, of size $w \times h \times 3$, the object CNN model CNN^{obj} and the scene CNN model $\text{CNN}^{\text{scene}}$, we compute the probability of sensitivity label $y_{sensitive}$. The convolution layers of both the models are followed by Global Average Pooling (GAP). For an input size $w \times h$ and F filters, the GAP layer outputs a vector of F dimensions which is the average of the filter activation of each location. The advantage of GAP is that the model learns all the features irrespective of the location. In particular, $f^{\text{obj}} = \text{GAP}(\text{CNN}^{\text{obj}}(I))$, $f^{\text{scene}} = \text{GAP}(\text{CNN}^{\text{scene}}(I))$. The combined image vector is $f^{\text{img}} = f^{\text{scene}} + f^{\text{obj}}$, which is the element wise addition of the object vector and the scene vector. The probability of sensitivity is computed as $P(y_{sensitive}) = sigm(W_I \times f^{\text{img}})$ where $W_I \in \mathbb{R}^{1 \times d^{\text{feats}}}$.

We resize the input image to 224×224. By fine-tuning the entire network, the model only keeps the features important for the task of image sensitivity.

Fusion Model. We combine the image model and the text model by concatenating the fixed sized vector outputs of the intermediate outputs of the two models. The text model and image model are pre-trained on our level 1 dataset and out images dataset respectively. After training the two models individually, the intermediate layer outputs contain useful features that are useful for sensitivity classification. Therefore,

we first pre-train the individual models and then concatenate the intermediate outputs to make the fusion model. The concatenated vector is fed to a FC layer with one output and sigmoid activation to get the sensitivity label. While training the fusion model, we freeze all the weights except for the last fully connected layer. Therefore, the model learns the features from the individual models which are useful, jointly. In particular, given a tweet represented as $\{I, \{w_1, w_2, ..., w_N\}\}$, the model predicts the probability of sensitivity label $y_{sensitive}$. The multimodal feature vector $f = [f^{\text{img}}; h_T]$, where f^{img} is the feature vector from the image model and h_T is the output of the GRU at the last step. For the tweets with no image we use a blank placeholder matrix, with zero values. The sensitivity probability is computed as $P(y_{sensitive}) = sigm(W_m \times f)$. All the hyperparameters were found by doing a grid search using a separate validation set.

C. Multi-Level Classifier

As most of the tweets in a random stream are not sensitive, a small false positive rate we would yield a large number of false positives compared to true positives. To solve this problem, we use multi-level classifiers. The first level of classification is done to remove the tweets which are not sensitive. Usually, these tweets belong to topics such as sports, entertainment, ads. After removing such tweets, we are left with tweets related to politics, news or other current events which have a high percentage of sensitive tweets. The level 1 classifier is also used to identify hashtags that contain sensitive tweets. The filtered tweets are then classified as sensitive or nonsensitive by the second level classifier. For level 1 classifier, we only use the text model trained on the level 1 dataset. For level 2 classifier we use the fusion model which takes both the text and the image as the input. The level 1 classifier is a weak classifier and trained on large amounts of weakly labeled data. The level 2 classifier is a precise classifier and trained on manually annotated data. Given a corpus of tweets, we first pass each tweet through the level 1 classifier. The tweets which are marked non sensitive by the level 1 classifier are discarded and the other tweets are passed to the level 2 classifier which decides the final sensitivity of the tweet.

V. Experiments and Results

In this section, we show the qualitative and quantitative performance of our models. We start by first explaining state-of-the-art baselines for content analysis in both textual and visual domains. Additionally, we conduct a deep inspection of the text and image models. We use class activation maps [23] to see what each classifier is actually learning. Finally, we end this section with a qualitative assessment by first responders to understand the real-world efficacy of our system.

A. Baselines

Feature Based Models. As the first text baseline we combine linguistic features and train an SVM classifier to classify the tweets. We start with a bag of words vector where we use top K frequent words of our dataset. We concatenate the vector with other features such as number of named entities and LIWC (Linguistic Inquiry and Word Count) [19] features. We also use Opinionfinder [28] for opinion mining to find whether the tweet is an opinion or factual. We also use events and topics [9] of the tweets as additional features.

Text Baselines. As discussed in Section II, we use Wulczyn et al. text toxicity detection [29] to test our model against. We call this the ToxicBaseline. For the text classification, we report the results on several popular text classification models. For deep learning models, we compare character level GRU, character level LSTM , word level GRU and word level LSTM. We also experiment by changing the number of hidden units and number and number of RNN layers.

Image baselines. For image classification, our baseline model extracts a feature vector from a pre-trained VGG16 model and use a simple classifier such as Logistic Regression and SVM on top to detect the sensitivity.

B. Quantitative Analysis

Evaluation Protocol. To evaluate the level 1 classifier, we do a testing and training split at the hashtags level. The tweets of the train hashtags are used for training, and the tweets of the testing hashtags are used for evaluation. We do that to ensure that the model is not getting biased by the hashtags. To evaluate the level 2 classifier, we use five-fold cross validation where we split the testing and training at tweet level since we have less annotated data for level 2 classifier. We use F1 score, accuracy, precision, and recall as evaluation metrics.

Results and Findings. Table V-B compares the different models for text classification and image classification. We see that the Recurrent Neural Network models are performing better than the SVM baseline because they capture the semantics of the text in an end to end manner.The word level model outperforms the character level model as it does not have to learn the words using the characters. By fusing the models, we see an improvement as we increase the features and the system can make better decisions. The improvement is by a less factor as a lot of tweets in the dataset does not contain any images. In the level 1 classifier, the two-layer word-level LSTM is gives better results as we have much more data for the level 1 training. For level 2, as the amount of data is limited, the GRU model is preferred due to its simpler structure and fewer number of parameters. On the same test set, the accuracy, precision, recall and F1-Scores from the ToxicBaseline are 67.5, 60.4, 67.5 and 63.5 respectively. We see an improvement in accuracy of 12.63% and 16.63% regarding F1-score.

Image models. Training a Logistic Regression model on the features of the images extracted by a VGG16 model is a strong baseline. However, by training the model in an end to end manner along with some novel practices, we show an improvement in the performance. By simply fine-tuning the same VGG16 model with a custom fully connected layer on top shows poorer performance due to a large number of parameters in the model. We see an improvement by using GAP and fine-tuning the complete network. By fusing the

TABLE V
COMPARISON OF DIFFERENT CLASSIFICATION MODELS. PROPOSED METHOD PERFORMS BETTER THAN THE BASELINE RESULTS.

(a) Image Model :				
Method	F1 Score	Accuracy	Precision	Recall
VGG16 Features + SVM	0.8065	0.8069	0.8079	0.8069
VGG16 Features + LR	0.8193	0.8194	0.8195	0.8194
VGG16 Finetuning	0.5350	0.5500	0.5400	0.4900
Object Model	0.8343	0.8438	0.8359	0.8344
Object + Scene Model	**0.8547**	**0.8550**	**0.8599**	**0.8550**
(b) Level 1 Tweet Classifier :				
Method	F1 Score	Accuracy	Precision	Recall
Word Level LSTM	0.7284	0.7302	0.7368	0.7302
2 Layer Word LSTM	**0.7372**	**0.7385**	**0.7440**	**0.7386**
Word Level GRU	0.7356	0.7369	0.7418	0.7369
(c) Level 2 Tweet Classifier :				
Method	F1 Score	Accuracy	Precision	Recall
SVM Baseline	0.682	0.701	0.739	0.654
Character Level GRU	0.7180	0.7619	0.6959	0.7619
Word Level GRU	0.7760	0.7816	0.7737	0.7816
Image + Text Model	**0.8013**	**0.8051**	**0.7983**	**0.8090**

object model with scene model and training it end to end yields an improvement of 2% in F1 score.

C. Model Inspection

While our quantitative evaluation demonstrate the performance of our approach, we further inspect the working of the models. Class activation maps [23] have been previously used to understand the performance of image classifier. In this work, we extend this method for examining the performance of our text classifier as well. To see which parts of the input contribute to the sensitivity, we using gradient-based localization for creating the class activation maps of the image model [31]. As we used GAP in our image model, we can see the prediction at a particular location of the image by feeding the features of that location rather than average of all locations. We then create a heatmap based on the predictions at each location. Figure 5 shows the heatmap created based on the locations of the image that make the image sensitive.

For the text model, we generalize the class activation mapping technique. To compute the contribution of each word, we first compute the importance of each feature of the word embedding by taking the average gradient of that feature with respect to the output. The features with higher gradient contribute more to the output. Then the importance score of each word is computed by a weighted sum of its embedding vector where the weights are the importance score of the individual features. Figure 6 shows the importance scores of the words of the tweets. The color of the highlighted word depicts the sensitivity of that word in the sentence.

For the text model words such as ”killed”, ”boycott”, ”arms” are assigned a very high sensitivity score by the model. We see that words which are given higher score are mostly of negative sentiment with respect to the context. As most of the images of our dataset contains protests, the image model is focusing on group of people in the protest. In one example image, we see that the burning object is contributing to the sensitivity.

Fig. 5. Class Activation Maps of some sensitive images for our Image Model. The highlighted regions correspond to the areas contributing to the sensitivity.

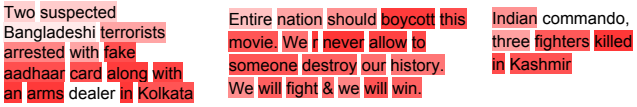

Fig. 6. Class Activation Maps of some text samples for our Text Model. The highlighted words contribute to the sensitivity of that sentence.

D. Qualitative Analysis

After establishing that our multimodal system theoretically outperforms state-of-the-art classification frameworks in both textual and image domains, we explore a quantitative analysis of the results. Therefore, to assess how well our system performs in a real-time setting we label 100 nonsensitive random tweets and 100 sensitive tweets with our classifier. Two annotators look at the scores given by our system and find 75 % to be correctly labeled. Detailed results of this evaluation are explained in Table VI.

TABLE VI
CONFUSION MATRIX FOR QUALITATIVE ANALYSIS CARRIED OUT BY TWO FIRST RESPONDERS. WE OBSERVE THE FALSE NEGATIVE RATE TO BE ALMOST NEGLIGIBLE.

	Labeled Positive	Labeled Negative
Positive	99	1
Negative	33	67

We note that there is only one false negative, implying that our system has a very low miss rate which is crucial for this use-case. Furthermore, to understand the false positives, we take a deeper look at them. We observe that 11 of the false positives have images that contain a crowd of people. This could be leading our classifier to get confused. 16 of the other false positives contain abusive words made sarcastically or mockingly. Since sarcasm detection is an ongoing research effort, we keep it out of the scope of our current study. We could not discern why our classifier was unable to discern why our classifier was confused by the remaining six tweets.

VI. DISCUSSION

In this work, we present a classification scheme for community moderation aimed at first responders. We do so by leveraging guidelines available on several web platforms bootstrapped by expert domain knowledge. We present a multilevel multimodal classifier which uses 1) a large corpus of weakly

labeled tweets 2) a smaller dataset annotated by first responders and 3) weakly labeled images to detect sensitive content.

Our system provides intelligence to first responders so that they may better tackle crises related content on social media. We emphasize our low false positive rate and posit that our system-labeled sensitive tweets can be used in cascade with a human-monitored system for further evaluation.

Limitations and Future Work. Our model inspects textual content as a first step of determining it's sensitivity, and therefore misses purely visual sensitive tweets or tweets which have sensitive images but not text. Currently our model makes a decision based on the image and textual dimension of the tweets. Network based features could also be important, as can be audio-visual compone. Additionally, CNNs and RNNs are computationally expensive, hence the inference would be slower. In addition to addressing these limitations, we plan to expand to other topics such as sexually explicit content.

REFERENCES

[1] Facebook Is Hiring 3,000 Moderators In Push To Curb Violent Videos, howpublished = https://www.forbes.com/sites/kathleenchaykowski/2017/05/03/facebook-is-hiring-3000-moderators-in-push-to-curb-violent-videos/#3804460758cb, note = Accessed: 2017-11-08.

[2] Moderators who had to view child abuse content sue Microsoft, claiming PTSD. https://www.theguardian.com/technology/2017/jan/11/microsoft-employees-child-abuse-lawsuit-ptsd. Accessed: 2017-11-08.

[3] Twitter Community Guidelines. https://support.twitter.com/articles/20175050. Accessed: 2017-11-08.

[4] Youtube Community Guidelines policies and safety. https://www.youtube.com/yt/about/policies/#community-guidelines. Accessed: 2017-11-08.

[5] Christina Boididou, Stuart E Middleton, Zhiwei Jin, Symeon Papadopoulos, Duc-Tien Dang-Nguyen, Giulia Boato, and Yiannis Kompatsiaris. Verifying information with multimedia content on twitter. *Multimedia Tools and Applications*, pages 1–27, 2017.

[6] Lauren Bacon Brengarth and Edin Mujkic. Web 2.0: How social media applications leverage nonprofit responses during a wildfire crisis. *Computers in Human Behavior*, 54:589–596, 2016.

[7] Stevie Chancellor, Yannis Kalantidis, Jessica A Pater, Munmun De Choudhury, and David A Shamma. Multimodal classification of moderated online pro-eating disorder content. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3213–3226. ACM, 2017.

[8] Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. Hate is not binary: Studying abusive behavior of# gamergate on twitter. *arXiv preprint arXiv:1705.03345*, 2017.

[9] Adrien Guille and Cécile Favre. Mention-anomaly-based event detection and tracking in twitter. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 375–382. IEEE, 2014.

[10] Andrew F Hayes and Klaus Krippendorff. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89, 2007.

[11] Homa Hosseinmardi, Amir Ghasemianlangroodi, Richard Han, Qin Lv, and Shivakant Mishra. Towards understanding cyberbullying behavior in a semi-anonymous social network. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 244–252. IEEE, 2014.

[12] Chi Yoon Jeong, Seung Wan Han, Su Gil Choi, and Taek Yong Nam. An objectionable image detection system based on region of interest. In *Image Processing, 2006 IEEE International Conference on*, pages 1477–1480. IEEE, 2006.

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

[14] Srijan Kumar, Robert West, and Jure Leskovec. Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In *Proceedings of the 25th international conference on World Wide Web*, pages 591–602. International World Wide Web Conferences Steering Committee, 2016.

[15] Sonia Livingstone. Taking risky opportunities in youthful content creation: teenagers' use of social networking sites for intimacy, privacy and self-expression. *New media & society*, 10(3):393–411, 2008.

[16] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI*, pages 3818–3824, 2016.

[17] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 708–717, 2017.

[18] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, pages 145–153. International World Wide Web Conferences Steering Committee, 2016.

[19] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

[20] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[21] Koustav Rudra, Siddhartha Banerjee, Niloy Ganguly, Pawan Goyal, Muhammad Imran, and Prasenjit Mitra. Summarizing situational tweets in crisis scenario. In *Proceedings of the 27th ACM Conference on Hypertext and Social Media*, pages 137–147. ACM, 2016.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[23] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization.

[24] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017.

[25] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[26] Limin Wang, Zhe Wang, Wenbin Du, and Yu Qiao. Object-scene convolutional neural networks for event recognition in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 30–35, 2015.

[27] William Warner and Julia Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics, 2012.

[28] Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Opinionfinder: A system for subjectivity analysis. In *Proceedings of hlt/emnlp on interactive demonstrations*, pages 34–35. Association for Computational Linguistics, 2005.

[29] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399. International World Wide Web Conferences Steering Committee, 2017.

[30] Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang. Cross-modality consistent regression for joint visual-textual sentiment analysis of social multimedia. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 13–22. ACM, 2016.

[31] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016.

[32] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.