Abusive Content Filtering Using Deep Learning

R. Srinivasan, Ankita Kumari*, Ashwin K. Ashok*

Department of Computer Science Engineering,
SRM Institute of Science and Technology, Kattankulathur
Chennai 603203, India
E-mail: srinivasan.r@ktr.srmuniv.ac.in
E-mail: ankitakumari_as@srmuniv.ac.in
E-mail: ashwink_kg@srmuniv.ac.in
*Corresponding Author

Abstract: Advancing technology has provided users with platforms to freely express themselves, but with every blessing there comes a curse. In past years, the use of abusive content in online content generated by the user has created a lot of issues for society. Hence, the detection of abusive content is increasingly gaining importance. This paper presents a basic model, which can be helpful for detecting such abusive content in comments, pictures, and GIFs. This model is a result of the fusion of concepts of Deep Learning for image processing and natural language processing which can be used for image description generation and hence allow for abusive content filtering with respect to an image. The training image helps to train our model to generate sentences closest to the target description. Finally, our model uses LSTM to detect if this caption generated, is abusive or not-abusive. This model is performing with the accuracy of 86.7%.

Keywords: Natural Language Processing, Long Short Term Memory Network, Convolutional Neural Network, Deep Learning, Abusive Content**.**

Biographical notes:

*This paper is a revised and expanded version of a paper entitled **[Abusive Content Filtering Using Deep Learning]** presented at [**International Conference of Internet of Things, SRM IST (Chennai, India) on 11.03.2019**].*

1 Introduction

Automatic image description generation for an image is a very intractable and demanding task, which could be of great help in identifying abusive content on the web or can be added as a new feature in social media applications that suggest a caption for our images. As a result of which, along with algorithms for better visual understanding of image we also need a natural language processing algorithm to express the semantic knowledge gained for the image into the understandable language (English). The description of the image should be generated in such a way that it should be able to describe all the objects, their attributes, action in which they are indulged and the relation between them. This generated description is then passed through a content moderation function to detect the presence of obscenity in our content. This paper proposes to use a deep convolutional neural network together with a Recurrent Neural Network (RNN) for the above-mentioned purpose. The Convolutional Neural Network (CNN) extract features from the input images. The proposed model does not train a classifier on top of extracted features, instead an RNN is trained to generate text, word by word. The error is the difference between the expected and generated text. This generated text is a representation of the image and its features. This generated text is then passed through a Long Short Term Memory (LSTM) network to detect the presence of obscenity in the generated text. The presence of obscenity in the text is a direct implication that the feature of an input image shows the presence of obscenity and hence the image has obscene content in it.

It took many attempts and proposals to move forward from the images to it's descriptions [1, 2]. In the proposed model, in place of attempting to join existing solutions we have developed a well constructed single model which takes the image or GIFs as input and tries to generate descriptions of the images very well and the words are in a perfectly sequenced manner. The proposed model has been trained to maintain the sequence of words to match the desired (target) sequence as much as possible.

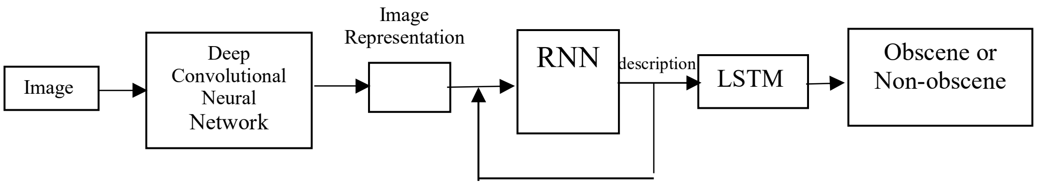

Figure 1. Content Filtering Model (CFM), proposed model. It consists CNN followed RNN for generating description of the input image. The description generated is classified by LSTM into obscene and not-obscene.

Some studies have shown that using the Recurrent Neural Networks (RNNs) [3,4,5] for translation of the source language (S) into a target language (T) can be done with great simplicity along with high accuracy. RNN is able to do so with the help of "encoder" and "decoder" RNNs. "Encoder" RNN takes in the source sentence and gives back fixed-length vector.

"Decoder" RNN treats the vector generated as an initial hidden state and uses the vector to generate target sentence. Over the course of time, the Convolutional Neural Network (CNN) has successfully replaced encoder RNNs. The CNN has satisfactorily shown the depiction of the input image in form of fixed length embedded vector. This depiction (fixed length vector) can be used for various vision-related purposes [6]. In Fig.1 we can see that CNN is used as "encoder", which has been pre-trained for classifying the images. RNN decoders take the last hidden layer of CNN as input and generate a description for the images. It is this description that is given to the LSTM model for detection of obscene content. We have named this model as Content Filtering Model (CFM).

Our contributions in this model are as follows: 1) We have included a module for image extraction so that GIFs can also be filtered. 2) This model has been carefully built from the combination of state-of-art subnetworks. The CFM has yielded the Bilingual Evaluation Understudy (BLEU) value in the range of 0.55 to 0.85 for image description generation module. The model is performing significantly well with an overall accuracy of 86.7% on the Flickr8k dataset.

2 Existing Models

Earlier, many studies have been done in the field of computer vision to solve the problem of description generation of videos [7, 9]. As a result of which, a combination of basic visual recognizer and structured formal languages like And-Or graphs or many logical systems came into existence. Rule-based systems convert the generated logical representation into a human understandable language (natural language). These systems were mostly hand-designed, hence they were relatively less robust and were prone to many faults, as a result of which they had limited applications.

In the last few years, there has been resurgence of interest in the problem of describing still images. Natural language generation systems are driven by further advances in object recognition. In some models [1] object detection and templates have been used to deduce the triplet of scene elements that were further transformed into text. Li et al. has [10] used a similar type of approach, to generate the final description. The detected objects and relationships between them were also taken into account.

In one model [2] more complex graphs were used to detect the set of triple scene elements. After that more powerful language-parsing based models were developed [11, 12, 13, 14, 15]. These models were successful in describing the images in the wild but still had rigidity in text generation.

Many approaches [16, 17, 18] suggested, ranking the descriptions for the given image. The idea was to embed the images and texts together in the same vector space. The similar type of images and descriptions were embedded in close spacial positions so that the descriptions that lie closer to the image in the embedding layer can be retrieved. The above approach, however failed to describe the combination of objects in an image, even though the image of objects, individually were given during training. These approaches also failed to evaluate how good the generated description was.

The work presented in this paper is a combination of the Convolutional Neural Networks for image classification [19] and Long Short Term Memory Neural Network for generation and classification of the description of an image. LSTM is a modified variation of RNN, which has solved the lack of memory problem and vanishing gradient problem. Followed by another LSTM layer to classify, if the generated description is obscene or not obscene.

Recent success in sequence generation in machine translation [3, 4, 5] has greatly inspired this model. The only difference is that instead of giving sentences at the beginning, we provide the model with features of the image that are extracted by CNN. Kiros et al. [21] have implemented such a model but with a feed-forward network. For the prediction of the next word, the previous word and the image were given as input. Feedforward networks are not very good at the prediction of the next word because it only considers the current input and has no notion of order in time. Later, RNN has been used by Mao et al. [22] in this prediction task. At last, Kiros et al. [23] with the help of a computer vision model and LSTM for natural language processing constructed a joint multimodal embedding space. When compared to our approach, instead of the single pathway they have used two paths, one for text that defines a joint embedding and another for images.

The proposed model uses more robust RNN, which directly takes visual data as input. These minute changes have allowed the proposed model to achieves better accuracy in the generation of descriptions.

Text classification has proven to be an important task and effective text classification has found several applications like spam filtering, sentiment analysis, language classification etc. The understanding of semantics and sentence structure still seems to be a hurdle and needed to be tackled more effectively. With improvement in deep learning approaches and development of new architectures using neural networks, the problem of semantic understanding has been taken care of to a certain extent.

Earlier many methods and classifiers have been applied, and each of them had their own advantage and disadvantages. Some classifiers were Naïve Bayes classifier [28, 29] which was simple, fast, had linear computation time and could be used for multi-label classification but it was assumed to be non-dependent on features. Next was KNN Classifier [30, 31], which was effective with large training data, was simple and non-parametric but had high computation time, difficulty in determining the correct type of distance measures and also gave less accuracy due to noisy features. Another classifier was Regression classifier. It was very versatile, worked well with linear relation, efficiently described the label and it's features and could be regularized to avoid overfitting. Regression classifier was not very effective with non- linear relation and training stage was expensive. Unlike other classifier SVM Classifier [32, 33] was able to solve the problem of non-linear relations. It also showed robustness against overfitting, was good with high-dimensional spaces and worked equally well with both large and small data sets. Some of its disadvantages were that it was memory intensive, hard to tune, wasn't effective in scaling. Apart from these, another classifier used was Decision tree classifier. [34, 35, 36] It used a hierarchical subset of the dataset, was easy to interpret, Nonparametric and performed well even with large datasets but had prominent overfitting problem. At last, Neural Network based classifier was used, it could learn non-linear and highly complex relationships, could create good generalizations over unseen data and could learn hidden relationships in the data. There were several architectures that were suited for a specific task like image classification and text analysis. Like all other classifiers, it also had some disadvantages such as training phase is computationally expensive, it required a enormous amount of training data, and was time-consuming.

Neural networks like Convolutional Neural Networks (CNN) have proven to be highly effective with images. Similarly, Recurrent Neural Network (RNN) has been very effective with sequential data like text documents. In a work [26] the authors discussed the way of using both RNN and CNN for text classification. This method tried to make use of the advantages of both the neural networks for developing a text classifier and it worked better when compared to traditional classifiers. The reason behind the betterment was the use of CNN to analyse the most relevant components of the text prior to text classification. In [27] the authors tried to enhance the existing recurrent neural network by enabling the neural network to adapt to long term dependencies. This was established using Long short-term memory (LSTM) units. This method has proven to be exceptionally good for text classification and other tasks related to natural language processing.

The neural networks were an excellent technique for classification tasks, however, it only suffers from the disadvantage of long training phases and the need for good sample data.

3 Proposed Model

The proposed model [Fig.2] is divided into three modules- 1) Feature Extractor, 2) Image Description Generator, 3) Text Classifier. The main aim of this model is to be able to filter GIFs, and for that we need to extract the still images from Gifs. The extracted images are then given to a pre-trained model to interpret the content or features of the images. We are using the VGG model (Oxford Visual Geometry Group), the winner of the ImageNet competition, 2014. VGG is just a variant of CNN. VGG consists of 16 layers, connected to each other using weights. To reduce the number of parameters it uses the 3x3 filter in each layer with stride size 1. The features of the image are given to an LSTM layer to produce the description corresponding to the features found in the image. As part of our image description generation model, the LSTM cell has four inputs- image feature, caption, mask, and current position. The model then uses Adams stochastic gradient descent and backpropagation to update weights. To generate the image description, we have applied softmax to get the word with the highest probability, then this recently generated image description along with the other inputs is fed back into the model. This process continues until the whole sentence is generated. The generated description is then further passed to another LSTM based classification module to detect the presence of abusive or obscene content in the text which is a direct indication of the presence or absence of abusive content in the image.

3.1 Convolutional Neural Net-based Feature Extractor

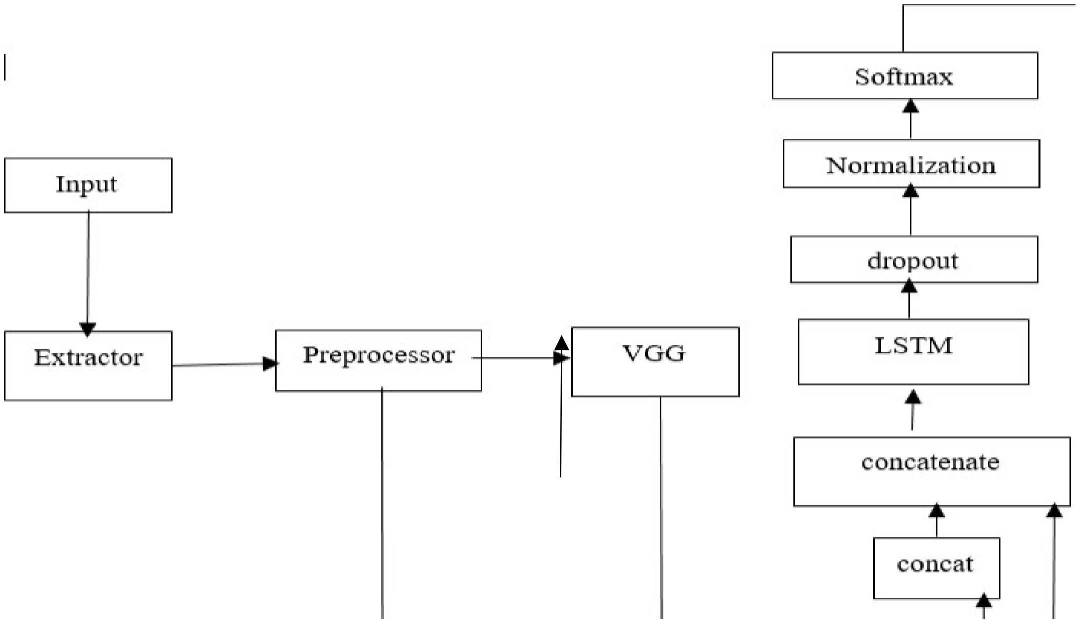

Figure 2. Architecture Diagram of proposed

CNN or ConvNet is an artificial neural network that has been so far used popularly for analysing images. Apart from image analysis, CNN can be used for classification problem and data analysis. The CNN is an artificial neural network which specializes in identifying or detecting patterns and make interpretation from them. This pattern detection ability is what makes CNN so effective for image analysis. A CNN consists of many hidden layers known as convolutional layers, these layers take in the input, then modifies the input in some way and then output the modified input to the next layer. This modification operation is known as a convolution operation. The number of filters that each convolutional layer has should be defined. These filters present in convolutional layer detect the patterns present in images. The pattern could be edges, corners, etc. Some filter may detect edges (vertical, horizontal, diagonal), some may detect corners, some may detect texture, etc. As the depth of the network increases the filters become more and more sophisticated. In deeper layers instead of just edges and simple shapes, our filter might be able to identify definite objects like handle, lights or windows and in further layers, the filters are capable of detecting even more sophisticated and advanced objects like a car or truck as a whole.

The end layer of the VGG-16 model is removed, as if implemented fully it will output the category of the image to which it belongs but for the CFM (Content Filtering Model) the only concern is the patterns recognized from the image just before the classification is made. These patterns are called the "features" of the image which can be extracted by the model.

In this model, image descriptions are generated using a combined neural network and probabilistic framework. The aim is to maximize the probability of the correct or accurate translation when the input sentence is given in an end-to-end manner.

With the help of RNN, variable length input is encoded into a fixed dimensional vector. The representation is then “decoded” to get the required output sentence. The maximized probability of getting the right description when given an input image can be calculated as:

$$\theta^* = \arg\max_{\theta} \sum_{(I,s)} \log p(S|I;\theta) \qquad (1)$$

Where, *θ = parameters of the model*
I = image
S = correct description

Chain rule it applied over S, so the joint probability is:

$$\log p(s|I) = \sum_{t=0}^{N} \log\ p(S_t|I, S_0, \ldots\ldots, S_{t-1}) \qquad (2)$$

Where *N = length of a particular example.*

We can see that in the above formula, the dependency on *θ* has been eliminated for ease (S, I) is taken as a training example. The optimization of the summation of log probability (eqn. 2) over the complete training set s done by Stochastic Gradient Descent (SGD) is. It is natural to model *p(St|I, S0,..., St−1)* with a Recurrent Neural Network (RNN), where the variable number of words we condition upon up to *t−1* is expressed by a fixed length hidden state or $h_{t.}$ Memory is revised using non-linear function whenever there is a new input $x_{t.}$

$$h_{t+1} = f(h_t, x_t) \qquad (3)$$

Where, h_t = *fixed length hidden state;*

To increase the efficiency of RNN, important design choices should be made like, what should be the form and type of *f* and how should be the images or word used as inputs $x_{t.}$ We have used LSTM net (long-short-term-memory) as *f*. Till now, LSTM has proven to be the best with sequence tasks such as translation. For image feature extraction, we are using CNN, which is the current state-of-the-art in pattern and object detection and recognition. With the means of transfer learning, CNN has also achieved success in scene classification [24]. In CNN representation of the words is done with the help of the embedded model.

To increase the efficiency of RNN, important design choices should be made like, what should be the form and type of *f* and how should be the images or word used as inputs $x_{t.}$ We have used LSTM net (long-short-term-memory) as *f*. Till now, LSTM has proven to be the best with sequence tasks such as translation. For image feature extraction, we are using CNN, which is the current state-of-the-art in pattern and object detection and recognition. With the means of transfer learning, CNN has also achieved success in scene classification [24]. In CNN representation of the words is done with the help of the embedded model.

3.2 Long Short Term Memory -based Sentence Generator

Most common challenges that are faced during designing and training RNNs are vanishing and exploding gradients [20]. Hence, function *f* should be wisely chosen to deal with these challenges. LSTM, a variant of RNN is introduced to check the mentioned challenges. When applied properly it showed high efficiency and huge success in translation [3, 5] and generation of sequence [25].

The main essence of LSTM is the memory cell (*c*) which encodes the knowledge associated with every input until this step [Figure 3.] LSTM model consists of "gates" to regulate the cell behavior. It consists of three gates – input, output and forget gate. Input gate *(i)* decides that whether or not to allow new input in, forget gate (*f*) delete information that is not needed anymore. Output gate (*o*) decides the impact of information on the current time step. The gates are analog, in the form of sigmoid, ranging from 0 to 1. Being analog enables it to do backpropagation for optimization.

The definition of cell updates, output and input gates are:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}) \quad (4)$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fm}m_{t-1}\right) \quad (5)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}) \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot h(N_{cx}x_t + W_{cm}m_{t-1}) \quad (7)$$

$$m_t = o_t \odot c_t \quad (8)$$

$$p_{t+1} = Softmax(m_t) \quad (9)$$

Where, $\odot$ = *product with the gate,*
W matrices = trained parameters,
σ(·) = sigmoid non-linearity
h(·) = hyper-tangent non-linearity
p_t = probability distribution over all words.

These multiplicative gates address the challenge of vanishing and exploding gradient and make the training process of LSTM more efficient and robust. Softmax is fed with m_t and it produces probability distribution p_t over all the words.

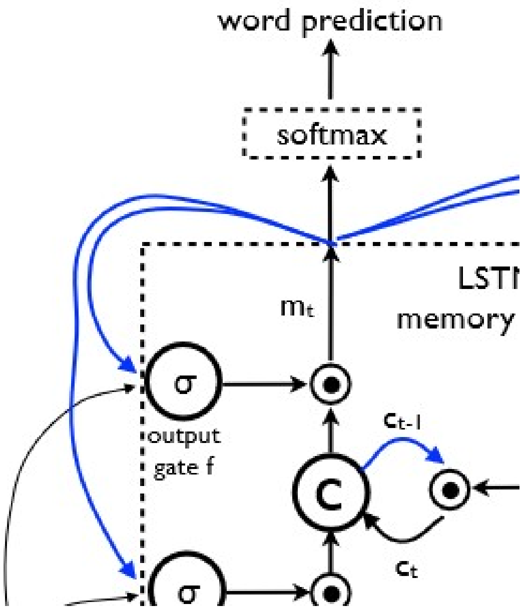

Figure 3. LSTM: The memory block contains a cell *c* which is controlled by the three gates. In blue we show the recurrent connections – the output *m* at time *t-1* is fed back to the memory at time *t* via the three gates; the cell value is fed back via the forget gate; the predicted word at time *t-1* is fed back in addition to memory output *m* at the time *t* into the Softmax for word prediction.

As part of our image description generation model, the LSTM cell has four inputs- image feature, caption, mask, and current position. The model then uses Adams stochastic gradient descent and backpropagation to update weights. To generate the image description, we have applied softmax to get the word with the highest probability, then this recently generated image description along with the other inputs is fed back into the model. This process continues until the whole sentence is generated.

After the LSTM model has gone through all the images and preceding words, it becomes fit for training. With objective to get a better idea of the training procedure, it is better to imagine the model in unrolled-form. Unrolling of the model means that a copies of LSTM memory are created for each word of an image such that all the LSTMs have the exactly same parameters.

In the unrolled version, these recurrent networks tend to behave like feed-forward connections. Suppose, the input image is donated as *I* and $S = (S_0, S_1, \ldots.., S_N)$ is used to denote a true sentence that describes this image. The unrolling-procedure of the model is as follows:

$$x_{-1} = Cnn(I) \tag{10}$$

$$x_t = W_e S_t, \quad t \epsilon \{0 \ldots N \quad 1\} \tag{11}$$

$$p_{t+1} = LSTM(x_t), \quad t\epsilon\{0 \ldots N - 1\} \tag{12}$$

Each word is represented as a one-hot encoded vector, S_t. S_t has a dimension equal to the size of the dictionary. Some special tags are used to denote the starting (S_0) and ending

(S_N) of the sentence. S_N shows that the complete caption has been generated. Word embedding, *We* layer maps the image and the words in the same space. The loss can be written as negative of the summation of the log-likelihood (probability of correct word over all other words) of the correct word at each-and-every step:

$$L(I,S) = -\sum_{t=1}^{N} \log p_t(S_t) \qquad (13)$$

The above-mentioned loss can be minimized in terms of all the parameters of the LSTM and word embedding *We*. along with the parameters of the first layer of image embedder convolutional neural network.

3.3 LSTM-based Sentence Classifier

In this model, LSTM has been used twice. First, it was used for description generation for images and second, for classifying these generated descriptions into two categories- Obscene or Not-Obscene. This section of the paper deals with the details on LSTM-based sentence classifier. It consists of two components: (i) a word embedding that maps each word in a sentence into lower dimension word vector; (ii) LSTM neural network which sequence and produces the word-level feature.

The first component of sentence classifier is the word embedding layer. It maps each-and-every word in the sentence to low-dimensional dense word vector. The word vector is then passed to the LSTM layer for processing. The embedding layer can be denoted as follows:

$$e_z = W_{em} x_t \qquad (14)$$

Where x_t = One-hot representation of *t-th* word v_t and $x_t \in \{0,1\}^{|v|}$;

|V| = Size of the vocabulary V;

W_{em} = stored the representation of all words in

$W_{em} \in R^{|V|x|D|}$;

The second component is the LSTM layer. The LSTM is just another variant of the recurrent neural network. Like RNN, LSTM also uses hidden layers and previous information to assign memory to the network. Apart from that, LSTM has more control over the memory. It can manage the amount of the current input that is useful and needed for new memory and can also delete the information for memory which is no longer needed.

This fine control allows LSTM to develop a long term memory and helps to make predictions over relatively longer sequences. LSTM has successfully resolved some of the problems associated with RNNs like vanishing and exploding gradient problem. LSTM comprises of four sub-units – input-gate, output-gate, forget-gate and a candidate memory cell.

First, at time *t* value for i_t(input-value) and $\tilde{c}_t$ (candidate-value) will be calculated:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \qquad (15)$$

$$\tilde{c}_t = tanh(W_i x_t + U_i h_{t-1} + b_i) \qquad (16)$$

Next, calculate the value for f_t, the activation function of memory cells forget-gate at time t:

$$c_t - i_{t^*}\tilde{c} + f_{t^*}c_{t-1} \tag{17}$$

The output-gates and their outputs can be written as:

$$o_t = \sigma(W_0 x_t + U_i h_{t-1}) \tag{18}$$

$$h_t - o_{t^*}\, tanh(c_t) \tag{19}$$

After computing the output, or activation, the resulting sequence from the LSTM layer I.e. $(h_1, h_2, \ldots., h_T)$, Where *T* represents the length of the input sequence of word to the layer. The LSTM layer is then followed by the dropout-layer for regularization and Softmax to give the respective categorical probability value. Dropout layer protects the model from overfitting and helps it to generalize well.

4 Experimental Results and Datasets

4.1 Datasets

This model has used Flickr8K dataset [39] for training the model to generate description of images. It's a very diverse dataset having 8092 images consisting of pictures from 6 different Flickr categories. These
categories are i) wild_child , ii) dogs_in_action , iii) action_photography , iv) outdoor_activities , and v) Flickr-social. This dataset consists of five descriptions for each image, all of them are slightly different from each other. The average length of description (caption) in this dataset is 11.8 words. Out of all the images, 6000 images along with their caption has been used as training data, 1000 as validation data and 1092 as test data.

Another dataset used for this project has been taken from “Toxic Comment Classification Challenge” [40] conducted by Kaggle. It is used to train the model to classify the sentence into obscene or not-obscene. It consists of various comments from Wikipedia, labeled by humans on the basis of toxicity of comment. The types of toxicity are i) toxic, ii) severe_toxic, iii) obscene, iv) threat, v) insult, vi) identity_hate. It consists of 1,59,572 training data and 1,53,165 test data.
The sentence that belongs to any of the types of toxicity is categorized as obscene and the sentence that doesn’t belong to any of the toxicity is categorized as not-obscene.

4.2 Techniques of Evaluation

The proposed model uses Bilingual Evaluation Understudy (BLEU) technique for determining the quality of machine generated description by comparing it to the human generated descriptions of the image. The closer a machine translation is to a professional human translation , the better it is. BLEU value lies between 0 to 1. The more it is closer to 1, the more is the relevance with the human translation and the better is the performance of model. The formula for calculating BLEU-n value is:

$$P_n = \frac{\sum_{n-gram} count_{HT(n-gram)}}{\sum_{n-gram} count_{MT(n-gram)}} \tag{20}$$

Where*, HT = Human Translation;*

MT = Machine Translation;
n-gram = Number of consecutive words taken together;
count = Number of times n-grams present in a sentence;

Example 1: - HT1- A man is walking on the road.
HT2- A man is moving on the road.
MT- Man is on the road.

Let n = 2 i.e. we have to take 2 consecutive word at a time.

N-gram	$\boldsymbol{count_{MT}}$	$\boldsymbol{count_{HT}}$ **(max)**
Man is	1	1
Is on	1	0
On the	1	1
The road	1	1
total	4	3

So, BLEU-2 score using equ.20 is $\frac{3}{4}$ i.e. 0.75.

The proposed model consists of binary classification I.e. it only consists of two classifications, obscene and not-obscene. The quality of such binary classifiers are checked by calculating accuracy, precision, recall and f-score. There are only four possible outcomes for the proposed model:

TO = Sentence is actually obscene and was predicted obscene
FN = Sentence is actually obscene but was predicted non-obscene.
FO = Sentence is actually not-obscene but predicted as obscene.
TN = Sentence is actually not-obscene and predicted as not-obscene.

Accuracy can be given as follows:

$$Accuracy = \frac{TO + TN}{TN + FN + FO + TP} \qquad (20)$$

Formula for recall is as follows:

$$Recall = \frac{TO}{TO + FO} \qquad (21)$$

Precision is calculated by following formula:

$$Precision = \frac{TN}{TN + FO} \qquad (22)$$

F-score can be evaluated using equation 20, 21, 22 and as follows:

$$F - score = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \qquad (23)$$

4.3 Experiment and Result

The model has two primary modules, the first module generated image description and the second module is a LSTM based text classifier. These two modules are trained separately before they are combined to produce a single deep learning function. Here, the training and evaluation of the two modules can be considered as two tasks. Task-1 involves the training and evaluation of the image description generation model and Task-2 can be considered as the training and evaluation of text classification. For Task-1, after training of the image description generation model, the BLEU scores [8] obtained after evaluation is shown in the Table 1.

As evident by table 1, it can be stated that although the proposed model is less accurate when compared to other three models but on the contrary, it uses simpler algorithms and deploys relatively less complex architecture. This gives CFM (Content Filtering Model) an upper hand and make it more suited for description generation for images.

BLEU	**CFM**	**Mao** [22]	**Jia** [24]	**Xu** [26]
BLEU-1	0.579114	0.565	0.647	0.670
BLEU-2	0.344856	0.386	0.459	0.457
BLEU-3	0.252154	0.256	0.318	0.314
BLEU-4	0.131446	0.170	0.216	0.213

Table 1. BLEU scores of the Image description generation model.

For Task-2, we train the LSTM model for text classification on the "Toxic comment dataset" provided by Kaggle. After evaluation of the model, we get the following metrics corresponding to the model.

Model	Precision	Recall	F score
LSTM	0.86264	0.87202	0.86731

Table 2. Evaluation result of Task 2

5 Conclusion

This paper presents CFM (Content Filtering Model), which aims to filter online obscene content like GIFs along with images and comments. It is a neural network based model which uses Convolutional Neural Network for processing of images and extracting features from it. The extracted features are used by an RNN based model for the generation of human understandable image descriptions. Further, LSTM networks classify the image descriptions as obscene or not-obscene. Various experiments on image processing and relation classification on dataset have proven that methods used in our model are competitive with the state-of-the-art deep learning methods. This model is robust in terms of qualitative results and quantitative evaluation that is calculated using the BLEU technique [8].

In the future, this model can act as a base model for various social media platforms to classify GIFs along with images and comments.

Insights discussed in this paper encourages future work on the development of architecture and method which can provide more efficiency and accuracy in terms of better recognition of pattern from images, caption generation, and text classification.

REFERENCES

[1]. A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: "Generating sentences from images". In ECCV, 2010.

[2]. G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: "Understanding and generating simple image descriptions". In CVPR, 2011.

[3]. K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H.Schwenk, and Y.Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In EMNLP, 2014.

[4]. D. Bahdanau, K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to align and translate". arXiv:1409.0473, 2014.

[5]. I.Sutskever, O.Vinyals, and Q.V.Le. "Sequence to sequence learning with neural networks". In NIPS, 2014.

[6]. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y.LeCun. Overfeat: "Integrated recognition, localization, and detection using convolutional networks." ArXiv preprint arXiv:1312.6229, 2013.

[7]. R. Gerber and H.-H. Nagel. "Knowledge representation for the generation of quantified natural language descriptions of vehicle traffic in image sequences." In ICIP. IEEE, 1996.

[8]. "https://en.wikipedia.org/wiki/BLEU" :for information related to BLEU value.

[9]. B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu. I2t: "Image parsing to the text description." Proceedings of the IEEE, 98(8), 2010. [33] P.Young, A.Lai

[10]. S.Li, G. Kulkarni, T.L.Berg, A.C.Berg, and Y.Choi. "Composing simple image descriptions using web-scale n-grams." In Conference on Computational Natural Language Learning, 2011.

[11]. M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. C. Berg, K. Yamaguchi, T. L. Berg, K. Stratos, and H. D. III. Midge: "Generating image descriptions from computer vision detections." In EACL, 2012.

[12]. A. Aker and R. Gaizauskas. "Generating image descriptions using dependency relational patterns." In ACL, 2010.

[13]. P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y.Choi. "Collective generation of natural image descriptions." In ACL, 2012.

[14]. P. Kuznetsova, V. Ordonez, T. Berg, and Y. Choi. Treetalk: "Composition and compression of trees for image descriptions." ACL, 2(10), 2014.

[15]. D. Elliott and F. Keller. "Image description using visual dependency representations." In EMNLP, 2013.

[16]. M. Hodosh, P. Young, and J. Hockenmaier. "Framing image description as a ranking task: Data, models and evaluation metrics." JAIR, 47, 2013.

[17]. Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. "Improving image-sentence embeddings using large weakly annotated photo collections." In ECCV, 2014.

[18]. V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: "Describing images using 1 million captioned photographs." In NIPS, 2011.

[19]. S. Ioffe and C. Szegedy. Batch normalization: “Accelerating deep network training by reducing internal covariate shift.” In arXiv:1502.03167, 2015.

[20]. S. Hochreiterand J.Schmidhuber. “Long short-term memory. Neural Computation”, 9(8), 1997.

[21]. R.Kirosand R.Z.R.Salakhutdinov. “Multimodal neural language models.” In NIPS Deep Learning Workshop, 2013.

[22]. J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille. “Explain images with multimodal recurrent neural networks.” In arXiv:1410.1090, 2014.

[23]. R. Kiros, R. Salakhutdinov, and R. S. Zemel. “Unifying visual-semantic embeddings with multimodal neural language models.” In arXiv:1411.2539, 2014.

[24]. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: “A deep convolutional activation feature for generic visual recognition.” InICML,2014.

[25]. A. Grave. “Generating sequence with the recurrent neural networks.” arXiv: 1308.0850, 2013.

[26]. LAI, S.; XU, L.; LIU, K.; ZHAO, J.. Recurrent Convolutional Neural Networks for Text Classification. AAAI Conference on Artificial Intelligence**,** North America, Feb. 2015. Available at: https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9745/9552. Date accessed: 10 Oct. 2018.

[27]. s. Hochreiter, J. Schmidhuber, "Long short-term memory", Neural Compute., vol. 9, no. 8, pp. 1735-1780, 1997.

[28]. G. Aghila and others, “A survey of naive Bayes machine learning approach in text document 1003.1795, 2010**.**

[29]. P. Bai, and J. Li, “The improved Naïve Bayesian WEB text classification algorithm,” International Symposium on CNMT 2009, IEEE, pp. 1-4, 2009.

[30]. T. Dong, W.Cheng and W. Shang, “The research of KNN text categorization algorithm based on eager learning,” International Conference on ICICEE, IEEE, pp. 1120-1123, 2012

[31]. K. Shi, L. Li, H. Liu, J. He, N. Zhang, W. Song, “An improved KNN text classification algorithm based on density,” IEEE International Conference on CCIS, pp. 113-117, 2011.

[32]. L. Youwen, X. Shixiong, and Z.Yong, “A supervised local linear embedding based SVM text classification algorithm,” Sixth WISA 2009, IEEE, pp. 21-26, 2009.

[33]. Z. Wang, X. Sun and D. Zhang, “An optimal text categorization algorithm based on SVM," IEEE, vol. 3, pp. 2137-2140, 2006.

[34]. J. R. Quinlan, “Induction of decision trees,” in Machine learning, vol. 1(1), Springer, pp. 81-106, 1986.

[35]. G. S. Chanvan, S. Manjare, P. Hedge and A. Sankhe, “A Survey of Various Machine Learning Techniques for Text Classification,” in IJETT, vol. 15, pp. 288-292, 2014.

[36]. D. D. Lewis and M. Ringuette, “A comparison of two learning algorithms for text categorization,”, vol. 33, pp. 81-93, 1994.

[37]. Mikolov, Tomas, Martin Karatiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. "Recurrent neural network based language mode!. " Tn Tnterspeech, vo!' 2, p. 3. 2010.

[38]. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan, “Show and Tell: A Neural Image Caption Generator". IEEE, vol. 20 2016.

[39]. M. Hodosh, P. Young and J. Hockenmaier (2013) "Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics", Journal of Artificial Intelligence Research, Volume 47, pages 853-899.

[40]. "https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data".