Using text and image machine learning to classify and prevent cyberbullying Final report submitted by BT5153 Group 5

Mukeshwaran Baskaran (A0165086Y) [1] **Nanhai Zhong (A0231953E)** [1] **Ng Boon Khai (A0231863E)** [1]
Kuan Ju Lin (A0231904M) [1] **Chen Hong (A0231993X)** [1]

Abstract

Group 5 github code link:
https://github.com/typing212/Cyberbullying-Detection

1. Introduction

Cyberbullying, defined as the use of digital technology to inflict harm (Englander et al., 2017), is a pressing issue that needs to be addressed. Compounded by the fact that the on-going COVID-19 pandemic has paved the way for decreased face-to-face interactions in favour of online ones, it is now more urgent than ever to solve the cyberbullying problem. For example, on April 15th 2020, UNICEF issued a warning about the rise in cyberbullying cases during the pandemic (1). The threat of cyberbullying is corroborated by the following statistics: 36.5% of schooling-age children have been cyberbullied before, which led to side effects such as decreased academic performance, depression, or even suicidal thoughts (2). Furthermore, aggressors are often anonymous, making them difficult to be stopped by the relevant authorities.

1.1. Objective

Therefore, correct identification of cyberbullying over the internet is imperative as a first-step towards introducing effective intervention strategies to stem cyberbullying propagation. To this end, a machine learning approach is ideal for the task at hand. For example, sentiment analysis using natural language processing (NLP) (Wang et al., 2020). The aim of this project is to build machine learning models and train on the cyberbullying kaggle dataset (Wang et al., 2020; 3) to classify whether or not a sample is categorized as cyberbullying or not. Further, a novelty of the dataset over previous ones within the literature is that the labelled data provides finer detailed information about the type of cyberbullying actions. For example, samples could be labelled as age, ethnicity, gender, religion, or others category of cyberbullying. This multi-classification fine-grain gives rise to opportunities for relevant targeted intervention strategies or counselling (Wang et al., 2020), assuming the perpetrators could be identified.

Beyond building traditional machine learning models such as logistic regression (LR), support vector machine (SVM), k-nearest neighbors (k-NN), and naive-bayes (NB) to solve the classification problem, this project aims to investigate the use of advanced machine learning models such as long short term memory (LSTM), and Bi-directional Encoder Representations from Transformers (BERT). Furthermore, to build on the work of Ref. (Wang et al., 2020), it may be worthwhile to investigate data augmentation techniques such as lexical replacement, back translation, text surface transformation, random noise injection, instance crossover augmentation, and generative methods to generate more text data in order to better understand the impact of having more data on deep learning model performance. Next, given that cyberbullying takes the form of both text and image, to round off the cyberbullying detection problem, a convolutional neural network (CNN) was built to classify cyberbullying images. Finally, a short discussion on the methods of Ref. (Vishwamitra et al., 2021) is given, where using contextual factors (list) captured by [software list], the basic CNN classification model can be greatly improved with multimodality learning. Note that the two classification tasks (text and image) are distinct and separate from each other.

2. Data

In this project, two datasets are considered: The labelled kaggle dataset for cyberbullying text, as well as a second dataset, credited to Ref. (Vishwamitra et al., 2021), which comprises a novel comprehensive labelled cyberbullying image dataset.

[1]Business Analytics Centre, National University of Singapore, Singapore 119613, Singapore. Correspondence to: Mukeshwaran Baskaran <e0157319@u.nus.edu>.

	religion	age	gender	ethnicity	not_cyberbullying	other_cyberbullying
# of records	7998	7992	7973	7961	7945	7823

Table 1. Number of records in each class

2.1. Text dataset

The textual dataset was obtained from Kaggle (2), which was originated from Ref. (Wang et al., 2020). The Kaggle dataset used in this section has 47692 rows and 2 features (tweet_text and cyberbullying_type). The initial exploratory data analysis shows that: Tweet text and cyberbullying type are the columns. Both columns are of the datatype "object" and are strings. The dataset has no missing values. The classes for cyberbullying types are relatively balanced, as shown in Tab. 1.

In addition, as demonstrated in the wordcloud displayed in Fig. 1, offensive tweets based on ethnicity have the most characters, followed by gender and religion-related offensive tweets.

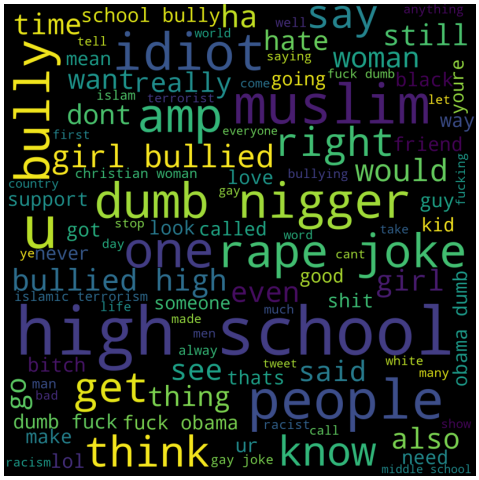

Figure 1. Word cloud visualization of all the tweets showing the most commonly occurring words.

The term "bullying" is the most common among the top 50 words in the "not-cyberbullying" category. This could indicate that the purpose of these tweets is to convey an anti-bullying message. The most popular tweets for gender are "rape," "gay," and "funny," meaning that gays are more likely to be the target of rape jokes on Twitter. The most popular word for religion is "muslim," implying that the bulk of hostile religious tweets are directed against Muslims. The N-word is one of the most commonly used words in the ethnicity category, implying that ethnicity-based slurs on Twitter are primarily directed at the black community.

2.2. Image dataset

The classes of dataset are imbalanced but acceptable, with 14,650 images belonging to the non-cyberbullying class, and 5,201 belonging to the cyberbullying class.

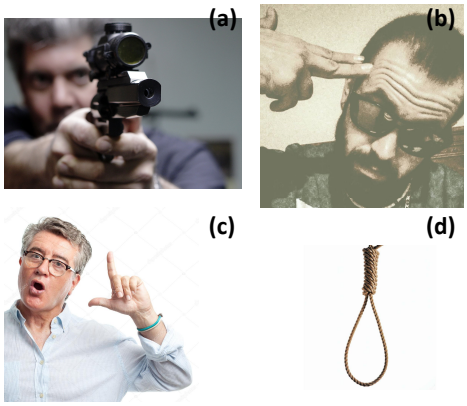

Figure 2. Sample examples of cyberbullying images.

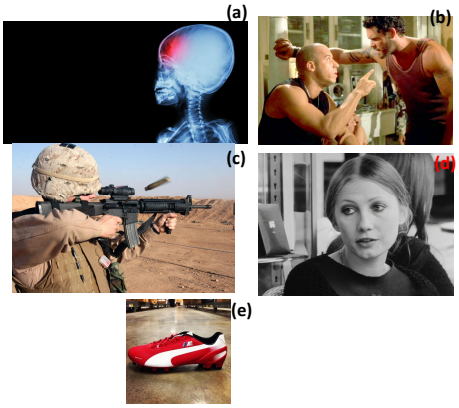

Figure 3. Sample examples of non-cyberbullying images.

Figs. 2 and 3 show sample examples of the cyberbullying image dataset (Vishwamitra et al., 2021) respectively for the class labels of cyberbullying, and non-cyberbullying.

According to Ref. (Vishwamitra et al., 2021), the difficulty of classifying image cyberbullying is that it is contextual. For example, in Fig. 2, objects such as guns (Fig. 2(a)), or hanging rope (Fig. 2(d)) are associated with cyberbullying as they suggest physical harm of some sort. On the other hand, in Fig. 3(c), an image of a soldier firing a gun is shown, but this image should not be a cyberbullying image as the soldier is just doing his job. Other types of objects such as a football shoe as shown in Fig. 3(e) should also not be associated with cyberbullying. Likewise, for the skull image in Fig. 3(a), it should not be a cyberbullying image as there is no contextual information suggesting harm to the victims.

Furthermore, the facial expression of a person are also important contextual cues in cyberbullying, as shown in the left figure Figs. 2(b) and (c). Aggressive facial expressions may be associated with cyberbullying. On the other hand, in Fig. 3(b), although the facial expression of the two persons are quite intense, they are not associated with cyberbullying. This is likely due to the fact that their body posture are not faced towards the front. All of the image cyberbullying classification are contextual. For example, in Fig. 3(d), although the woman's body posture is facing to the front, her expression is calm, so it is also not a cyberbullying image.

A discussion on the important contextual factors in image data, and how to extract them, will be discussed further in Sec. 7.2.

3. Text data preprocessing

Before feature engineering, several preprocessing steps were conducted to clean the text data. To convert raw strings of text to encodings which can be consumed by a machine learning model, the raw text needs to be treated with a series of preprocessing steps to filter and clean the text data. Next, use NLTK (Natural Language Toolkit), WordNetLemmatizer, demoji packages to achieve this. The steps include:

- tokenization,
- lowercasing,
- stop-words filtering,
- lemmatization/stemming,
- removing special characters, punctuations, and digits,
- removing hashtag, mention mark and URLs,
- spell checking,
- converting emoji into words.

It is important to be mindful that some preprocessing steps may cause loss in information especially in a social media context and therefore need to be chosen wisely. For instance, in a tweet symbol "#" is usually used as hashtags and symbol "@" is used to refer to a twitter handle. Some authors may also deliberately misspell a word or use internet slang to express certain sentiments.

The next step is transforming the unstructured text data into structured data so that classification models could be built. In this project, word-level encoding, Word2Vector and BERT were used. Word-level encoding includes Bag-of-Words and Term Frequency-Inverse Document Frequency, which are the most prevalent encoding methods for converting text sentences into numeric vectors for statistical models. In essence, BoW is a count of word occurrences, whereas TF-IDF encoding also takes into account the importance of the words. Word2Vector considers whether those words have similar semantics when encoding. BERT is a word embedding model based on the self-attention mechanism that is pre-trained on top of the bidirectional transformer.

3.1. For statistical language models

Probabilistic models that learn the probability distribution of words are known as statistical language models. For statistical language models, both BoW and TF-IDF were used as encoding methods and establish classification models based on Naïve Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM) and Decision Tree (DT). To begin with, the cleaned text data was split into training dataset and test dataset. The ratio between the number of records in the training dataset and the test dataset is 3:1. Next, use the CountVectorizer function and TfidfTransformer function to calculate the BoW and TF-IDF of each word. To avoid data leakage, both are fitted on the training dataset and then transform for both training dataset and test dataset.

3.2. For neural language models

Classic statistical models perform and generalize less well than neural language models. Word embeddings are used as inputs to a neural network, which converts a sentence's words into vectorized representations. In the vector space, words with similar semantics have comparable representations. Word2Vector and BERT are some of the most often used word embedding technologies.

3.2.1. LSTM WITH WORD2VECTOR

To use Word2Vector encoding, a vocabulary of the top 5000 words from the training dataset was created. Next, tokenize the text using the vocabulary, add padding to tokens to keep them in the same length, and create a 200-dimension word embedding matrix by Word2Vector. As an example of the top 20 most common words in the vocabulary, see Fig. 4.

Fig. 4 shows that there may be a lot of text related to

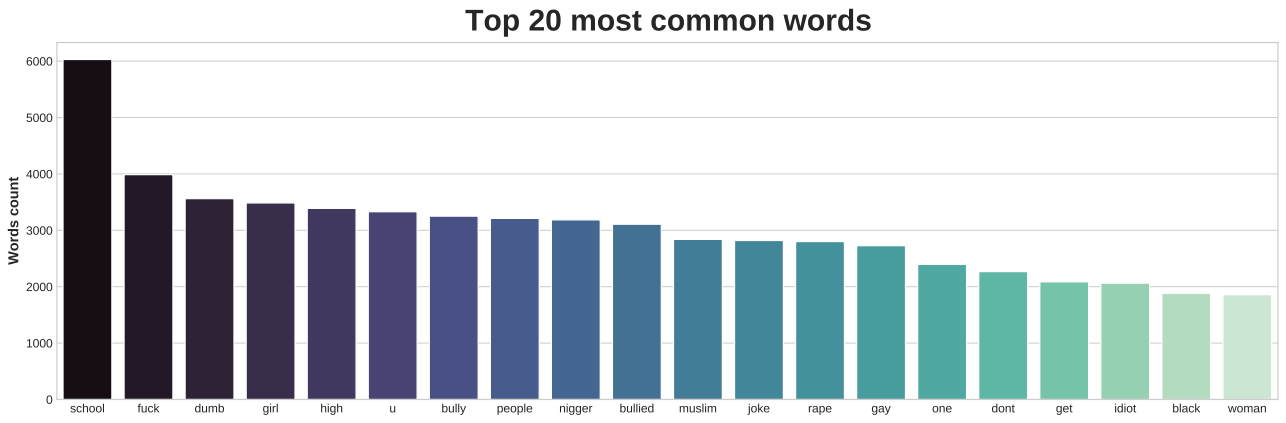

Figure 4. Top 20 Most Common Words in the Vocabulary

"school", which is consistent with the fact that school is a place where bullying might occur. In addition, some words represent a group of people are quite common, such as "girl", "muslim" and so on. Some swear words also have a high frequency. After creating the word embedding matrix, further split the original training dataset into training dataset and validation dataset. Besides, random over sampling to the training dataset was performed, so that each class has 4823 records.

3.2.2. BASIC BERT

A pre-trained BERT model from the Hugging Face library was used to perform classification. BERT is a context-based language model which is designed to be bidirectionally trained on transformer architecture. After data encoding, a Neural Network with 1 hidden layer, 50 hidden nodes and ReLU activation function was built.

3.2.3. ROBERTA

Next, a variant of BERT, RoBERTa, which is a more efficiently pre-trained model was experimented. The motivation of this variant is to improve upon basic BERT which is significantly undertrained. The central idea was to train the same BERT model for longer (more epochs) and on more data.

4. Text data results

4.1. For statistical language models

The out-of-sample performance of basic classification models are shown respectively in Tabs. 2 and 3 for accuracy and recall as follows.

Tab. 2 shows that there is no big difference of accuracy between different encoding methods. Among different basic classification models the Logistic Regression and SVM outperform the Naïve Bayes and Decision Tree. However, the training process of the SVM model is very time consuming compared with the Logistic Regression model.

	NB	LR	SVM	DT
BoW	0.74	0.81	0.81	0.78
TF-IDF	0.73	0.81	0.81	0.78

Table 2. Accuracy of Different Model Combinations

Table 3 shows that the classes with clear themes are easier to be detected by those models compared with not_cyberbullying and other_cyberbullying. Among those classes with a clear theme, the recall of gender is only around 80% while recalls of other classes are higher than 90%. In addition, although accuracy of different encoding methods is similar, the recalls with different encoding methods are different. When it comes to recall, the Naïve Bayes model with BoW has better performance than the Naïve Bayes model with TF-IDF, while the Decision Tree model with TF-IDF performs better than the Decision Tree model with BoW.

4.2. For neural language models

4.2.1. LSTM WITH WORD2VECTOR

LSTM represents Long Short-Term Memory, which is a recurrent neural network (RNN) extension that allows it to learn meaningful context over longer sequences. The hyperparameters of the bidirectional LSTM is shown in Tab. 4. A linear layer and a softmax layer were added at the end of the Bi-LSTIM model to transform the output into 6 dimensions (6 classes) and range between 0 to 1. The in-sample accuracy of the Bi-LSTM model was found to be 93%, while the out-of-sample accuracy was 80%, which

Class	NB BoW	NB TFIDF	diff	LR BoW	LR TFIDF	diff
not_cb	0.3	0.3	0	0.52	0.55	0.03
gender	0.8	0.8	0	0.82	0.8	-0.02
religion	0.98	0.97	-0.01	0.93	0.95	0.02
other_cb	0.42	0.37	-0.05	0.64	0.6	-0.04
age	0.99	0.98	-0.01	0.97	0.97	0
ethnicity	0.96	0.93	-0.03	0.98	0.98	0
Class	SVM BoW	SVM TFIDF	diff	DT BoW	DT TFIDF	diff
not_cb	0.46	0.52	0.06	0.48	0.46	-0.02
gender	0.8	0.79	-0.01	0.81	0.82	0.01
religion	0.93	0.93	0	0.91	0.92	0.01
other_cb	0.75	0.64	-0.11	0.52	0.56	0.04
age	0.96	0.97	0.01	0.96	0.96	0
ethnicity	0.98	0.98	0	0.97	0.97	0

Table 3. Recall of Different Model Combinations, where not_cb stands for not_cyberbullying, and other_cb means other_cyberbullying

Hyperparameters	
num_classes	6
hidden_dim	100
lstm_layers	1
learning_rate	3.00E-04
dropout	0.5
epochs	5

Table 4. The Hyperparameters of the Bi-LSTM model

indicates the model is not overfitting.

4.2.2. Basic BERT

The out-of-sample accuracy is the highest among previous models reaching 84%. At the same time, gender and not_cyberbullying achieve 87% and 55% recall respectively, surpassing the previous models, while others still have high recall.

4.2.3. Roberta

The out-of-sample accuracy of RoBERTa surpasses BERT with a score of 87% accuracy, owing to its improved model capability to capture contextual meanings of words.

4.2.4. Performance comparison between BERT and Roberta

Comparing the classification accuracy performance of BERT and Roberta, the results are summarized in Tab. 5.

Class	LSTM with Word2Vector	Basic BERT	Roberta
not_cyberbullying	0.46	0.55	0.65
gender	0.82	0.87	0.89
religion	0.91	0.96	0.96
other_cyberbullying	0.71	0.68	0.86
age	0.94	0.97	0.98
ethnicity	0.97	0.98	0.98
Accuracy	**0.80**	**0.84**	**0.87**

Table 5. Performance (accuracy) Comparison Between BERT and Roberta

4.2.5. Text augmentation

Data augmentation is used to generate additional synthetic training data by applying transformation to the existing training data. For natural language data, NLPAug library is used to implement various text augmentation techniques. The combination of the following text augmentation techniques are implemented to about 20% of train data:

- Synonym replacement via word embeddings, and
- Back translation

RoBERTa is used as the benchmark to study the augmentation effects on model performance. The results are shown in Tab. 6.

a) Synonym replacement
Synonym replacement obtains different sentences with the same meaning by replacing certain words with their corresponding synonyms based on word embeddings.

b) Back translation
Back translation generates additional data with different wordings and sentence structure by translating the existing data to a different language and subsequently translating it back to the original language.

From Tab. 6, it is evident that text augmentation has little to no effect on the performance in this application. It is suspected that this could be due to loss in information that captures cyberbullying content during translation or synonym replacement. Terms and phrases that appear in cyberbullying text often contain profanity, defamation, and vulgar expressions that only hold contextual meanings in local language. Hence, any replacement of words or translation may not be effective to enrich the dataset as they may result in introducing text samples with entirely different meanings than the original ones.

Class	Roberta	Roberta with T.A.	diff
not_cyberbullying	0.65	0.59	0.07
gender	0.89	0.91	-0.02
religion	0.96	0.97	-0.01
other_cyberbullying	0.86	0.78	0.08
age	0.98	0.98	0
ethnicity	0.98	0.98	0
Accuracy	**0.87**	**0.87**	**0**

Table 6. Comparing performance (accuracy) before and after text augmentation, where Roberta with T.A. indicates Roberta with text augmentation.

5. Image data preprocessing

After reading in the data, reshaping and scaling images was conducted, all images were resized to 100×100 pixels, so as to make sure the CNN model can process the data smoothly. Thereafter, the data was randomly split with 80 percent to be for training and 20 percent to be for testing.

6. Image data results

6.1. CNN model

Convolutional Neural Network is a type of neural network model which is well known for working with the images and videos, CNN takes the image's raw pixel data, trains the model, then extracts the features automatically for better classification. In the CNN model used, a series of convolution layer (ConV2D), followed by a Max pooling and a Dropout layer were built in the first part of deep learning model. Convolutional layers apply a convolution operation to the input, passing the result to the next layer. A convolution converts all the pixels in its receptive field into a single value. Here, the most common type 2D convolution layer was applied. It is a filter that slides over the 2D input data, performing element wise multiplication and sums up the results into a single pixel output. Max pooling is a pooling operation that selects the maximum value of elements from the region of the feature map where filter covers. It helps to reduce the dimensionality, and thus reduces the number of parameters to learn. In addition, it also removes noise from input data and retains only the significant values. The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. After the first series of layers, a flatten layer was connected between the following Dense layer and the previous ones to flatten the multi-dimensional input tensors into a single dimension. Several Dense and Dropout layers were constructed afterwards.

Summarily, the CNN network architecture is displayed in Fig. 9 in **Appendix**.

6.2. Evaluation

For evaluation, the ROC-AUC and accuracy are metrics used to evaluate the CNN classifier model for cyberbullying images prediction.

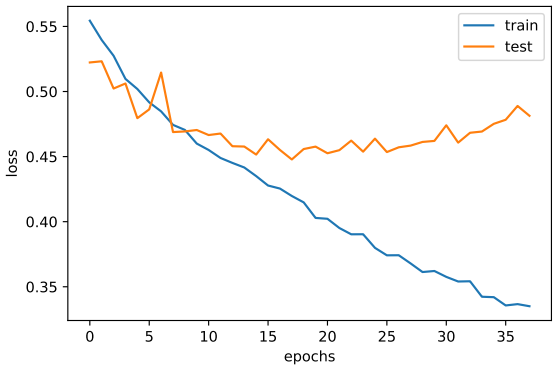

Figure 5. Training and testing loss plotted against number of epochs.

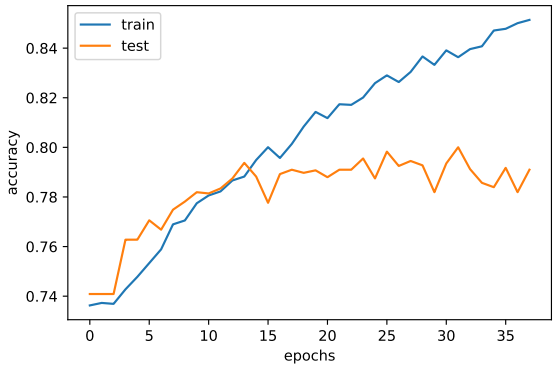

Figure 6. Training and testing accuracy plotted against number of epochs.

During training, the Epochs was set to be 50, with callback monitoring the validation loss for early stopping and storing the best parameters. As shown in Fig. 5, the training loss decreases as the epochs increases. However, when it reaches to epoch equals around 15, it appears to begin overfitting the data, since testing loss start to increase at that moment. In Fig. 6, the accuracy performance of the CNN model is illustrated. The trained CNN model can reach 79% accuracy on the testing data (assuming class separation criteria Z=0.5), which outperforms the naive baseline model if it is assumed that the naive model predicts every data point to be non-cyberbullying class.

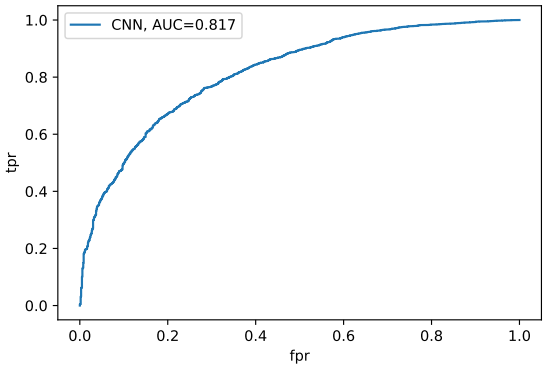

Figure 7. ROC curve, with AUC score of 0.817.

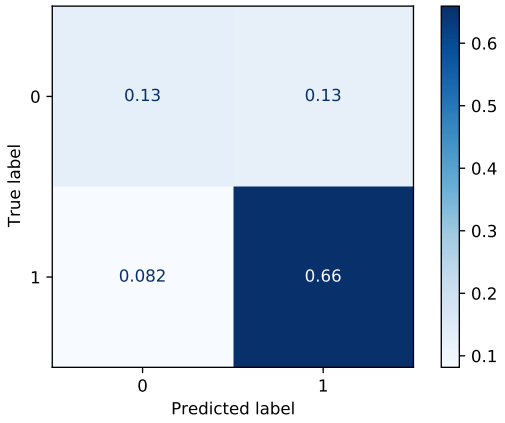

Figure 8. Confusion matrix of CNN classifier, assuming class separation crtieria Z=0.5.

The ROC-AUC and confusion metrics are also important evaluation metrics to consider. The ROC analysis provides a means of reviewing the performance of a model in terms of the trade-off between False Positive Rate and True Positive Rate (as class separation criteria Z is varied). As shown in Fig. 7, the CNN model reaches 0.817 in terms of AUC score. Fig. 8 displays the confusion matrix (for Z=0.5), with precision equals 0.617, and recall equals 0.508, indicating a reasonable classifier performance.

7. Discussion

7.1. Project application

Youngsters tend to spend most of their free time on social media platforms like TikTok and Twitter. These platforms have had tremendous difficulty in moderating the content shared by their users. The classification models can be utilized by social media platforms like Twitter and TikTok to reduce the exposure of harmful cyberbullying content towards children. The NLP models could be used by twitter to ensure users below a certain age do not get to view explicit content while TikTok could protect their underage users with the computer vision model.

7.2. Improvement to image cyberbullying classification: multimodality model

While the basic baseline CNN model performs adequately, achieving validation accuracy of 0.79, and recall of 0.508, there is still room for improvement. Note that for the CNN model, the training image and class labels were fed to the CNN model without doing anything else. In other words, the CNN model may not have adequately learned the contextual information of cyberbullying images.

In the research paper of Ref. (Vishwamitra et al., 2021), the novelty of their contribution lies in being the first of its kind to 1), not only provide a comprehensive cyberbullying image dataset, but 2), to contextualise the factors involved in cyberbullying images. This subsection discusses the approach taken by the authors to do image factor extraction to build a multi-modality learning model together with CNN to achieve a high out-of-sample classification accuracy of 93.36%.

Mainly, there are 5 contextual factors to consider: Body-pose, facial emotion, hand gesture, objects, and social factors such as anti–LGBT or hate speech related factors. For body pose, the authors used OpenPose (Cao et al., 2017) to extract where the persons involved in images are facing. By doing cosine similarity analysis of the body pose occurrences in cyberbullying and non-cyberbullying images, they found that body posture that directly faces the front are more likely to be associated with cyberbullying, while non-front facing postures are more likely be to associated with non-cyberbullying.

In facial emotion factor extraction, the authors used OpenFace (Baltrušaitis et al., 2016) to extract the emotions of the persons involved in images. Surprisingly, the cosine similarity analysis revealed that cyberbullying images tend not to show strong emotions. Counter intuitively, cyberbullying images subjects tend to show happy emotions, perhaps as a way to mock the victims.

In hand gesture factor extraction, they used Google Cloud Vision API (5). Hand gestures such as loser, middle finger, thumbs down, or gun point are associated with cyberbullying images.

For object factor extraction, the authors used YOLO (Redmon & Farhadi, 2018), you only look once object detection algorithm. Although majority of cyberbullying images do not have any objects, some portion of them contain threaten-

ing objects such as gun or knife to intimidate victims. The object factor is also an important feature.

Finally, for social factors such as hate speech or anti LGBT factors, for example "black face" or "hanging rope", it is difficult currently to do factor extraction. For this factor, the authors manually analysed the images and hand label these factors.

Finally, with these factors obtained, the authors built a multimodality model that combines the images of the image dataset with the features of the extracted feature. With this ML model, they achieved a mean accuracy of 93.36

7.3. Conclusion

In conclusion, in this project, the problem of machine learning classification of cyberbullying was addressed, both for the text data, and the image data separately. For the text data, traditional approaches such as statistical language models, for example bag of words, tfidf, and using these features to train classifical classifiers such as svm, lr, dt, and naive bayes was considered. These classifiers performed adequately, achieving average accuracy of about 74-81%. Next, for text data classification, advanced models such as neural language models were also considered. For neural language models, the models built were LSTM with word2vec, BERT, and Roberta. As expected, the deep learning models performed better, achieving average accuracy of about 80-87%. Since deep learning models perform best when there is a lot of data, text augmentation techniques such as back translation, and synonym replacement were experimented. It was found that text augmentation, when applied to Roberta, had little to no effect in terms of performance.

The second part of the project involves image cyberbullying classification. For this problem, the authors of Ref. (Vishwamitra et al., 2021) were contacted directly in order to gain access to their novel comprehensive labelled dataset. Next, a CNN model was built to perform the classification task and achieved validation accuracy of 0.79, in agreement with the baseline CNN model performance stated in Ref. (Vishwamitra et al., 2021). Further, in Sec. 7.2, the various contextual factor information such as body-pose, facial expression, hand gesture, objects, and social factors which could be used as features to build an effective multimodality machine learning model were discussed. Due to time limitation and technical difficulty of the factor extraction, it was not possible to experiment with this model.

Finally, it has to be said that cyberbullying is an important problem to tackle. Since more people are on social media these days, and victims of cyberbullyings, especially children, are reported to suffer from reduced academic performance, or even worse still, struggle with suicidal thoughts, preventing cyberbullying is an important problem to tackle. It is hoped that the work of this project contributes to solving this problem and, in turn, brings society one small step closer towards a better world with less suffering.

Appendix

Appendix contains only the CNN network architecture, Fig. 9.

References

(1) https://www.unicef.org/press-releases/children-increased-risk-harm-online-during-global-covid-19-pandemic.

(2) https://www.kaggle.com/andrewmvd/cyber bullying-classification.

(3) http://www.broadbandsearch.net/blog/cy ber-bullying-statistics.

(5) google cloud vision api, 2020. https://cloud.google.com/vision/.

Baltrušaitis, T., Robinson, P., and Morency, L.-P. Openface: an open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–10. IEEE, 2016.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7291–7299, 2017.

Englander, E., Donnerstein, E., Kowalski, R., Lin, C. A., and Parti, K. Defining cyberbullying. *Pediatrics*, 140 (Supplement_2):S148–S151, 2017.

Redmon, J. and Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Vishwamitra, N., Hu, H., Luo, F., and Cheng, L. Towards understanding and detecting cyberbullying in real-world images. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2021.

Wang, J., Fu, K., and Lu, C.-T. Sosnet: A graph convolutional network approach to fine-grained cyberbullying detection. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 1699–1708. IEEE, 2020.

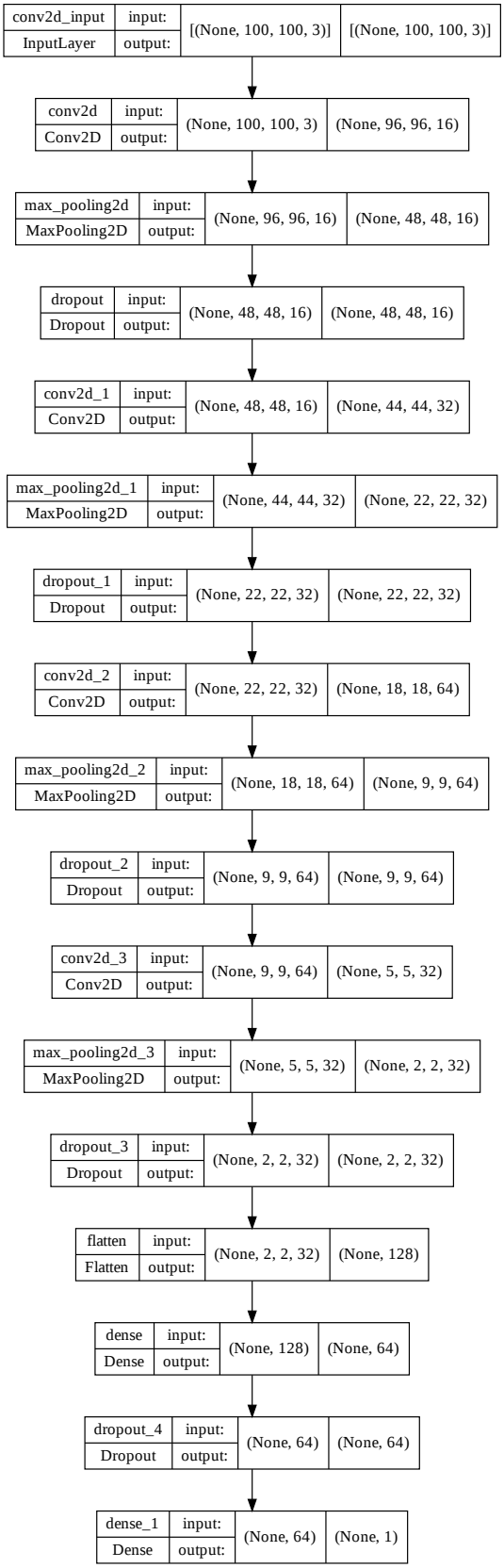

Figure 9. The network architecture of CNN used for the image cyberbullying classfication problem.