JSCE (Journal of System and Computer Engineering)
Vol.7, No.3, July 2026, pp. 334-343
ISSN (online): 2723-1240
DOI : https://doi.org/10.61628/jsce.v7i3.2759

DNN vs CNN for YouTube Shorts Abusive Word Classification

Evan Febditya Pratama [1,a,*]; Ryan Ari Setyawan [2,b]; Jemmy Edwin Bororing [3,c]
[1,2,3] Department of Informatics, Faculty of Engineering, Janabadra University, Yogyakarta, Indonesia
[a] 22330053@janabadra.web.id; [b] ryan@janabadra.ac.id; [c] jemmy@janabadra.ac.id
** Corresponding author*

Abstract

Abusive words displayed in YouTube Shorts thumbnails can be difficult to detect because the meaning may depend on both the text and the visual content. This study compares a Deep Neural Network (DNN) and a Convolutional Neural Network (CNN) for binary abusive-word classification using thumbnail images and OCR-derived text. The dataset contained 1,013 thumbnails collected from 1,013 different YouTube Shorts videos using TubeTrendy from March to May 2026. Text was extracted from each thumbnail using EasyOCR, cleaned, and converted into a 2,500-dimensional TF-IDF representation. Initial labels were generated through keyword-based screening of the OCR results and then verified by one researcher using the thumbnail and OCR result. The dataset was divided using stratified random splitting into 658 training, 202 validation, and 153 testing samples. Since each video contributed only one thumbnail, samples from the same video could not appear in different subsets. The text vectorizer was fitted only on the training data. The CNN achieved an accuracy of 0.8627 and a macro F1-score of 0.8527, while the DNN achieved an accuracy of 0.8301 and a macro F1-score of 0.8195. Under the current model configurations, the CNN-based multimodal model achieved higher test performance than the DNN-based multimodal model.

Keywords*: Abusive word classification; YouTube Shorts; Deep Neural Network; Convolutional Neural Network; multimodal classification.*

1. Introduction

The rapid growth of social media and short-form video platforms has changed how people share information, opinions, and entertainment. However, this growth has also increased the spread of abusive and harmful expressions in online content. Abusive language refers to words or phrases that contain insults, vulgar expressions, or offensive meanings. These expressions may be used as jokes, emotional reactions, or direct attacks on other people (Ibrohim & Budi, 2018). In Indonesian online communication, abusive language is often written using informal spelling, slang, abbreviations, repeated characters, and modified words. These variations make automatic detection more difficult (Ibrohim & Budi, 2018, 2019, 2023).

Previous studies have examined abusive language and hate speech detection on several social media platforms. Ibrohim and Budi (2018, 2019) developed Indonesian datasets and classification methods for abusive language and hate speech detection on Twitter. Salim and Suhartono (2021) studied abusive language and hate speech in Indonesian YouTube comments using recurrent neural networks. These studies show that automatic detection has received growing attention in Indonesia. However, most previous studies focus mainly on text from tweets, comments, or posts.

Short-form video platforms also contain harmful expressions in visual content. YouTube Shorts thumbnails may include text overlays, screenshots, captions, or other written elements. In this type of content, the abusive meaning may be related to both the displayed text and the visual

structure of the thumbnail. Therefore, text-only analysis may not fully represent the information contained in the image.

Multimodal machine learning combines information from more than one data type, such as text, images, audio, or video (Baltrušaitis et al., 2019). In harmful-content detection, multimodal approaches can be useful because offensive meaning may appear through the relationship between visual and textual information. Previous studies have applied multimodal methods to harmful memes, text-embedded images, and video content (Gomez et al., 2020; Kiela et al., 2020; Suryawanshi et al., 2020). More recent studies have also shown the importance of combining text and image features for detecting harmful or hateful content in social media (Bhandari et al., 2023; El-Sayed & Nasr, 2024; Ganguly et al., 2024; Thapa et al., 2022, 2023, 2024).

Deep learning is suitable for this task because it can learn complex patterns from high-dimensional data (Janiesch et al., 2021). A Deep Neural Network (DNN) can process image and text features through fully connected layers. However, when an image is flattened into a one-dimensional vector, some spatial relationships between pixels may be reduced. A Convolutional Neural Network (CNN) can preserve and learn local spatial patterns through convolutional operations (Krizhevsky et al., 2012). This ability may be useful for YouTube Shorts thumbnails because written words and visual elements can appear in different positions, sizes, and layouts.

Although multimodal harmful-content detection has been studied in memes, images, and videos (Boishakhi et al., 2021; Das et al., 2023; Prabhu & Seethalakshmi, 2025), direct comparison between DNN and CNN architectures for abusive-word classification in YouTube Shorts thumbnails remains limited. Existing Indonesian studies mainly focus on text-based content. Research that combines thumbnail images with OCR-derived text is still underexplored.

This study compares the performance of DNN and CNN models for binary abusive-word classification using 1,013 thumbnail and OCR-text pairs collected from 1,013 different YouTube Shorts videos. The textual input was extracted from each thumbnail using EasyOCR, while the visual input was obtained from the RGB thumbnail image. Initial labels were generated through keyword-based screening of the OCR results and then verified by one researcher using the original thumbnail and OCR result. The main objective was to determine whether the CNN model could achieve better classification performance than the DNN model on the same dataset. The results are expected to provide empirical information about the use of image and OCR-derived text features for abusive-word detection in YouTube Shorts thumbnails.

2. Method

This study used a quantitative experimental method to compare the performance of a Deep Neural Network (DNN) and a Convolutional Neural Network (CNN) for abusive-word classification. Each sample consisted of two inputs: a YouTube Shorts thumbnail and OCR-derived text extracted from the same thumbnail. Therefore, the task was treated as a multimodal binary classification problem.

The samples were classified into two categories: abusive and non-abusive. In the dataset, these categories were stored as kata_kasar and bukan_kasar. The two models were trained and evaluated using the same dataset split, text representation, initial image input size, optimization settings, and evaluation metrics. The main difference between the models was the method used to process the thumbnail images. The DNN used fully connected layers, while the CNN used convolutional layers to learn spatial image features.

This formulation is consistent with multimodal learning, which combines information from different data types to support a prediction task (Baltrušaitis et al., 2019). In this study, the image input represented the visual structure of the thumbnail, while the text input represented the words detected by EasyOCR. The comparison was designed to examine whether convolutional image processing produced better classification performance than fully connected image processing

2.1 Dataset and Preprocessing

The dataset was collected from March to May 2026 using TubeTrendy to obtain thumbnail images from YouTube Shorts. The collection was not limited to specific YouTube channels. Only one thumbnail was retained from each source video. Samples were included when the thumbnail had sufficient image quality and could be processed for text extraction. The final dataset contained 1,013 thumbnails obtained from 1,013 different YouTube Shorts videos.

Text was extracted from each thumbnail using EasyOCR. The initial OCR result was stored in the raw_ocr_text field. The extracted text was then cleaned by converting characters to lowercase, removing irrelevant symbols, and normalizing repeated spaces. The cleaned result was stored in the text field and used as the textual input of the models. The original thumbnail was also reviewed because OCR could incorrectly recognize, omit, or change some displayed words.

Initial labels were generated through keyword-based screening of the OCR results and then verified by one researcher using the original thumbnail and OCR result. A sample was assigned the final abusive label when the visible text contained an insulting, vulgar, degrading, or offensive Indonesian word or expression. A sample was labeled non-abusive when no such expression was identified. The researcher examined both the original thumbnail and the OCR result before assigning the final label. The OCR result supported the labeling process, but it did not automatically determine the class.

Because the labeling was performed by one researcher, there was no disagreement resolution process between annotators. Inter-annotator agreement was also not calculated. The use of one annotator is recognized as a limitation of this study.

The final dataset contained 633 non-abusive samples and 380 abusive samples. Each image file was checked by matching the filename in the CSV file with the corresponding file in the image folder. All 1,013 samples were retained because every metadata record had a valid thumbnail file. The labels were encoded into binary values, where bukan_kasar was encoded as 0 and kata_kasar was encoded as 1.

The thumbnail images were loaded in RGB format, resized to 96 by 96 pixels, and normalized to values between 0 and 1. The cleaned text was transformed using a TextVectorization layer with a maximum vocabulary size of 2,500 tokens and TF-IDF output.

The dataset was divided at the individual-sample level using stratified random splitting with a fixed random seed of 42. The stratified procedure preserved the class distribution in the training, validation, and testing subsets. Since each video contributed only one thumbnail, the sample-level split was equivalent to a source-video-level split. Therefore, thumbnails and OCR-derived text from the same video could not appear in different subsets.

The TextVectorization layer was fitted only on the training text. The fitted vectorizer was then applied to the validation and testing text without being fitted again. This procedure prevented vocabulary information from the validation and testing subsets from entering the training process. The final dataset distribution is shown in Table 1.

Table 1. Dataset distribution after splitting

Data Split	Non-abusive	Abusive	Total
Training	411	247	658
Validation	126	76	202
Testing	96	57	153
Total	633	380	1,013

2.2 Proposed Model Architecture

This study compared two multimodal models, namely a Deep Neural Network (DNN) and a Convolutional Neural Network (CNN). Both models received two inputs: a thumbnail image and a 2,500-dimensional TF-IDF text vector. The same text-processing branch was used in both models so that the main comparison focused on the image-processing method.

The text branch contained two dense layers with 80 and 40 neurons. Each dense layer used the ReLU activation function, L2 regularization, batch normalization, and dropout. The dropout rate for the text branch was set to 0.45. This branch produced a compact representation of the OCR-derived text.

In the DNN model, the thumbnail image was first received at a size of 96 by 96 pixels. The image was then resized to 32 by 32 pixels and flattened into a one-dimensional vector. The flattened vector was processed by two dense layers with 64 and 32 neurons. Each layer used ReLU activation, L2 regularization, batch normalization, and a dropout rate of 0.35. This process produced the visual representation used by the DNN.

In the CNN model, the 96 by 96 pixel thumbnail was processed through three convolutional blocks. The blocks used 16, 32, and 48 filters, respectively. Each block contained a 3 by 3 convolutional layer, batch normalization, max pooling, and dropout with a rate of 0.30. A global average pooling layer was then applied, followed by a dense layer with 48 neurons. This structure allowed the CNN to learn local spatial patterns from the thumbnail image, such as text position, shape, and layout (Krizhevsky et al., 2012).

For both models, the visual and textual representations were combined using a concatenation layer. The combined features were processed by two fusion layers with 64 and 32 neurons. The fusion layers used ReLU activation, L2 regularization, batch normalization, and dropout. The dropout rate in the fusion section was set to 0.50.

The final output layer contained one neuron with a sigmoid activation function. The output represented the probability that a sample belonged to the abusive class. A probability threshold of 0.50 was used to assign the final binary prediction. The data preparation process is shown in Figure 1, while the comparative multimodal modeling and evaluation stages are presented in Figure 2.

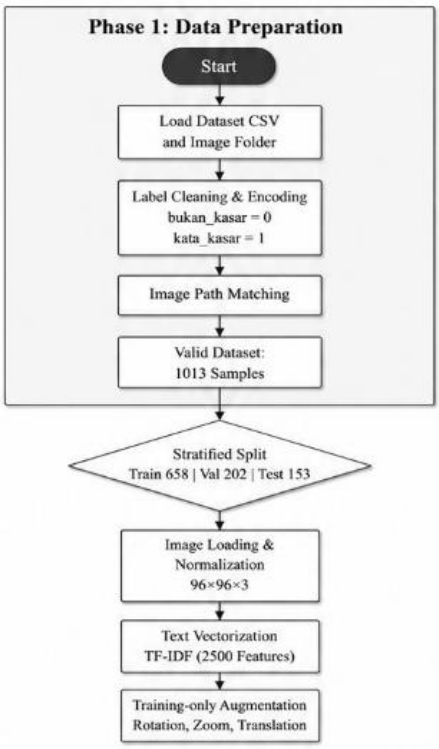

Figure 1. Data preparation workflow.

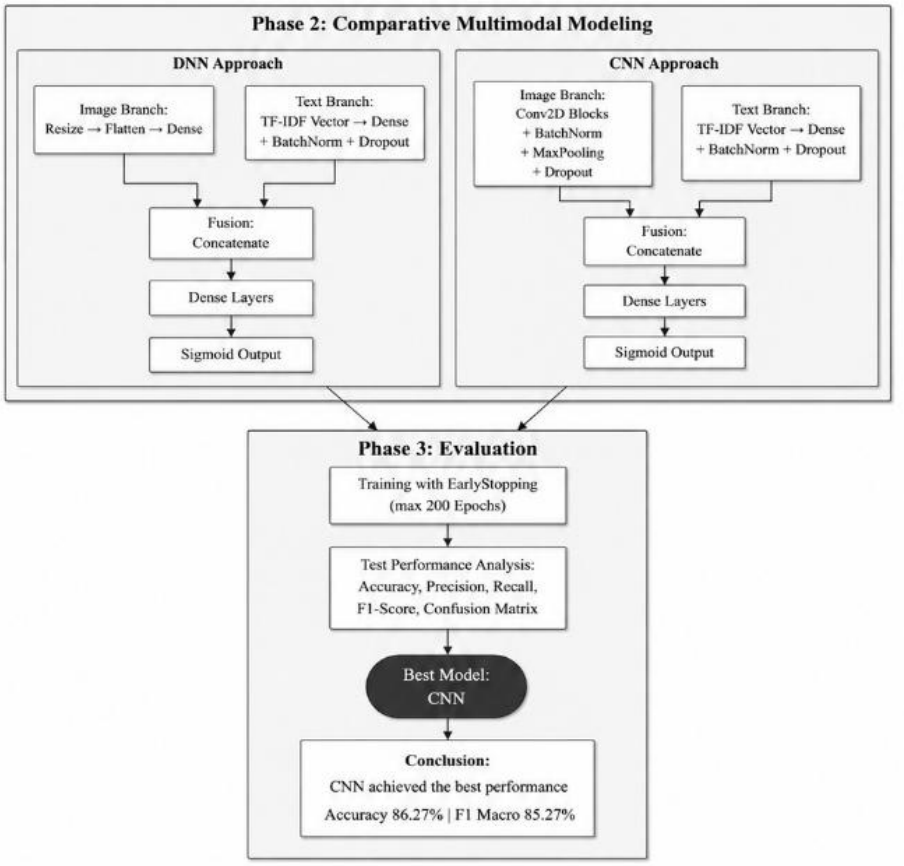

Figure 2. Comparative multimodal modeling and evaluation workflow.

2.3 Training and Evaluation Scenario

The DNN and CNN models were trained using the same main training settings. The Adam optimizer was used with an initial learning rate of 0.002 and gradient clipping with a maximum norm of 1.0. Binary cross-entropy was used as the loss function with label smoothing of 0.035. The maximum number of epochs was 200, and the batch size was 32. A fixed random seed of 42 was used to support reproducibility.

The learning rate was reduced when the validation loss did not improve for four consecutive epochs. The learning rate was reduced by a factor of 0.5, with a minimum value of 0.000001. Early stopping was also applied by monitoring the validation loss. Training stopped when the validation loss did not improve for ten consecutive epochs, and the model weights from the best validation result were restored.

Several methods were used to reduce overfitting. These methods included L2 regularization with a value of 0.001, batch normalization, dropout, and image augmentation. The dropout values differed between model branches, as described in Section 2.2. No class weights were used during training.

Image augmentation was applied only to the training subset. The augmentation included small rotation, zoom, and translation changes. The rotation factor was 0.005, the zoom factor was 0.015, and the horizontal and vertical translation factors were 0.01. The validation and testing images were not augmented. Therefore, model evaluation was carried out using the original processed thumbnails.

Model performance was evaluated using the testing subset of 153 samples. The evaluation metrics were accuracy, macro precision, macro recall, and macro F1-score. Accuracy measured the overall proportion of correct predictions. Macro precision, macro recall, and macro F1-score calculated the average performance across the abusive and non-abusive classes.

Macro F1-score was included because the dataset contained 633 non-abusive samples and 380 abusive samples. This metric provided a more balanced evaluation of both classes. A probability threshold of 0.50 was used to convert the sigmoid output into a binary prediction. Samples with a probability of 0.50 or higher were classified as abusive, while samples with a lower probability were classified as non-abusive.

3. Results And Discussion

The DNN and CNN models were evaluated using the same testing subset of 153 samples. The evaluation metrics were accuracy, macro precision, macro recall, and macro F1-score. These metrics were used because the dataset contained more non-abusive samples than abusive samples. Macro-averaged metrics gave equal importance to both classes and provided a more balanced view of model performance.

The training and validation curves are presented in Figure 3. These curves show the changes in accuracy and loss during the training process. The DNN completed training after 29 epochs, while the CNN completed training after 42 epochs. Early stopping ended the training process when the validation loss did not improve for ten consecutive epochs. The best model weights were then restored.

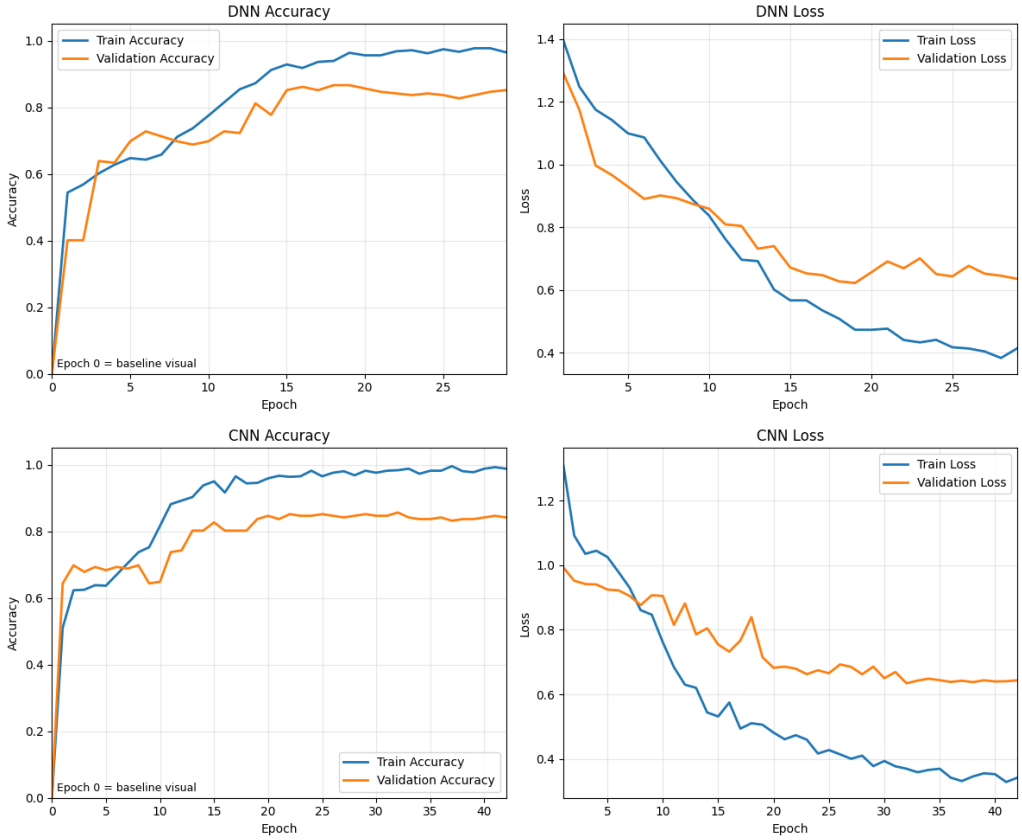

Figure 3. Training and validation accuracy and loss curves for the DNN and CNN models.

The DNN reached its lowest validation loss at epoch 19. The CNN reached its lowest validation loss at epoch 32. This result shows that the CNN required more epochs to reach its best validation performance. The longer training process may be related to the convolutional operations used to extract spatial features from the thumbnail images.

At the end of training, the DNN obtained a training accuracy of 0.9650 and a validation accuracy of 0.8515. The difference between the two values was 0.1136. The CNN obtained a training accuracy of 0.9878 and a validation accuracy of 0.8416, with a difference of 0.1463.

These differences show that both models performed better on the training data than on the validation data.

The gap between training and validation performance indicates a risk of overfitting. This condition is reasonable because the dataset contained only 1,013 samples, while both models had a large number of parameters. Early stopping, dropout, L2 regularization, batch normalization, and image augmentation were used to reduce the risk of excessive overfitting. However, this study did not compare models with and without these regularization methods. Therefore, the results only show that regularization was applied, not that each method independently improved generalization.

The test results are presented in Table 2. The DNN achieved an accuracy of 0.8301, a macro precision of 0.8175, a macro recall of 0.8218, and a macro F1-score of 0.8195. The CNN achieved an accuracy of 0.8627, a macro precision of 0.8540, a macro recall of 0.8514, and a macro F1-score of 0.8527.

Table 2. Performance comparison of DNN and CNN models

Model	Accuracy	Macro Precision	Macro Recall	Macro F1-score
DNN	0.8301	0.8175	0.8218	0.8195
CNN	0.8627	0.8540	0.8514	0.8527

The CNN achieved higher values than the DNN for all four evaluation metrics. The difference was 0.0326 for accuracy, 0.0365 for macro precision, 0.0296 for macro recall, and 0.0332 for macro F1-score. These results show that the CNN produced better overall and class-balanced performance on the current testing subset.

The visual comparison of the evaluation results is shown in Figure 4. The figure presents the accuracy, macro precision, macro recall, and macro F1-score obtained by both models

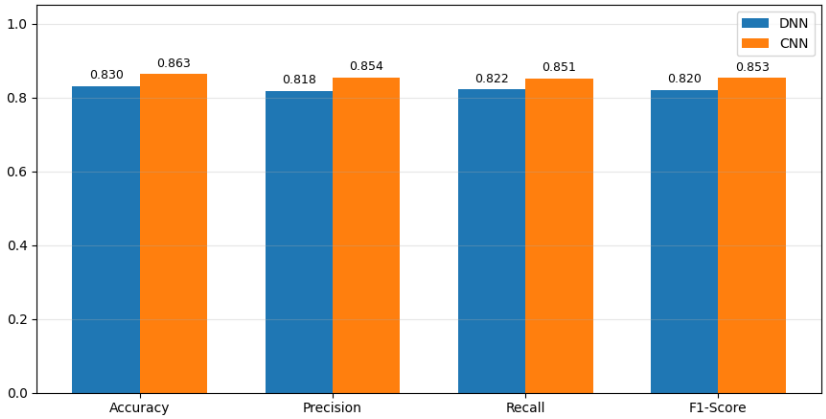

Figure 4. Performance comparison between DNN and CNN models.

The CNN also used fewer parameters than the DNN. The DNN contained 409,977 total parameters, including 409,417 trainable parameters and 560 non-trainable parameters. The CNN contained 233,753 total parameters, including 233,097 trainable parameters and 656 non-trainable parameters. Therefore, the CNN achieved higher test performance with a smaller total number of parameters.

The average training time per epoch was 2.8878 seconds for the DNN and 3.0730 seconds for the CNN. The CNN required slightly more time per epoch because convolutional operations were used to process the thumbnail images. However, the difference in training time was small in the current experimental setting.

The findings should be interpreted as a comparison within the current dataset and testing split. The study used one stratified train-validation-test division with a fixed random seed of 42. Therefore, the reported differences describe the results of this experiment and should not be interpreted as proof that the CNN will always outperform the DNN on other datasets or data splits.

The higher test performance of the CNN may be related to the way it processed the thumbnail images. Convolutional layers can learn local spatial patterns, such as the position, shape, and arrangement of text and visual elements. This ability is useful for thumbnail images because written words may appear in different locations, sizes, and layouts. In contrast, the DNN processed the image as a flattened vector. Flattening may reduce the spatial relationships between nearby pixels (Krizhevsky et al., 2012).

However, the difference between the two models should not be explained only by the use of convolutional layers. The DNN resized each thumbnail from 96 by 96 pixels to 32 by 32 pixels before flattening, while the CNN processed the thumbnail at 96 by 96 pixels. Therefore, the CNN received more spatial detail than the DNN. The difference in test performance may have been influenced by both the model architecture and the image resolution.

Both models used the same OCR-derived text branch and the same 2,500-dimensional TF-IDF representation. Therefore, the main experimental difference was found in the image-processing branch. The results show that the CNN-based image branch produced higher test metrics than the fully connected image branch under the current settings.

The study used both thumbnail images and OCR-derived text as model inputs. This design is consistent with multimodal learning, which combines information from more than one data type (Baltrušaitis et al., 2019). Previous studies have also used text and image information to detect harmful content in memes and social media images (Gomez et al., 2020; Kiela et al., 2020; Suryawanshi et al., 2020).

However, this study did not include text-only and image-only models. Therefore, the results cannot confirm how much each modality contributed to the final prediction. The findings only show that, when the same OCR-derived text branch was combined with two different image branches, the CNN model achieved higher performance than the DNN model. Future research should include unimodal baselines and ablation tests to measure the separate contribution of image and text features.

This study has several limitations. First, the dataset contained 1,013 samples, so the results may not represent the wider variety of abusive expressions found in YouTube Shorts thumbnails. Second, the samples were labeled by one researcher. Therefore, inter-annotator agreement could not be measured, and some context-dependent expressions may reflect individual judgment. Third, the original video URLs and video identifiers were not retained in the final dataset. As a result, the one-video-one-thumbnail procedure depended on the data collection process and could not be checked again from the final metadata.

The study also used one train-validation-test split with a fixed random seed of 42. Different data splits or random seeds may produce different results. In addition, the DNN processed thumbnails resized to 32 by 32 pixels, while the CNN processed thumbnails at 96 by 96 pixels. Therefore, the performance difference may have been affected by both the model structure and the image resolution. Finally, the study used only two labels, abusive and non-abusive. This binary structure did not represent differences in abusive targets, severity, context, or word variation.

Future studies should use larger datasets, more than one annotator, inter-annotator agreement measurement, repeated experiments with different random seeds, and source-video identifiers. They should also compare models using the same image resolution and include text-only, image-only, and multimodal baselines.

4. Conclusions

This study compared a Deep Neural Network (DNN) and a Convolutional Neural Network (CNN) for binary abusive-word classification using YouTube Shorts thumbnails and OCR-

derived text. The dataset contained 1,013 thumbnails collected from 1,013 different YouTube Shorts videos. Text was extracted using EasyOCR, and initial labels generated through keyword-based screening were verified by one researcher using the original thumbnails and OCR outputs. The CNN achieved an accuracy of 0.8627 and a macro F1-score of 0.8527, while the DNN achieved an accuracy of 0.8301 and a macro F1-score of 0.8195. Under the current model configurations, the CNN-based multimodal model achieved higher test performance than the DNN-based multimodal model. The higher performance may be related to the CNN's ability to learn spatial patterns from thumbnail images. However, the CNN also processed images at a higher resolution than the DNN, so the difference cannot be explained only by the model architecture. The findings are also limited by the dataset size, the use of one annotator, one train-validation-test split, and the absence of text-only and image-only baselines. Future studies should use larger datasets, multiple annotators, repeated experiments with different random seeds, equal image resolutions, and unimodal baseline models to provide stronger evidence about the contribution of each input type.

References

Baltrušaitis, T., Ahuja, C., & Morency, L. P. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 41*(2), 423–443. https://doi.org/10.1109/TPAMI.2018.2798607

Bhandari, A., Shah, S. B., Thapa, S., Naseem, U., & Nasim, M. (2023). CrisisHateMM: Multimodal analysis of directed and undirected hate speech in text-embedded images from Russia-Ukraine conflict. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 1994–2003). IEEE. https://doi.org/10.1109/CVPRW59228.2023.00193

Boishakhi, F. T., Shill, P. C., & Alam, M. G. R. (2021). Multi-modal hate speech detection using machine learning. In *2021 IEEE International Conference on Big Data (Big Data)* (pp. 4496–4499). IEEE.

Chhabra, A., & Vishwakarma, D. K. (2023). A literature survey on multimodal and multilingual automatic hate speech identification. *Multimedia Systems, 29*, 1203–1230. https://doi.org/10.1007/s00530-023-01051-8

Das, M., Raj, R., Saha, P., Mathew, B., Gupta, M., & Mukherjee, A. (2023). HateMM: A multi-modal dataset for hate video classification. *Proceedings of the International AAAI Conference on Web and Social Media, 17*(1), 1014–1023. https://doi.org/10.1609/icwsm.v17i1.22209

El-Sayed, A., & Nasr, O. (2024). AAST-NLP at multimodal hate speech event detection 2024: A multimodal approach for classification of text-embedded images based on CLIP and BERT-based models. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)* (pp. 139–144). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.case-1.19

Ganguly, A., Bin Emran, A. N., Puspo, S. S. C., Raihan, M. N., Goswami, D., & Zampieri, M. (2024). MasonPerplexity at multimodal hate speech event detection 2024: Hate speech and target detection using transformer ensembles. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)* (pp. 125–131). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.case-1.17

Gomez, R., Gibert, J., Gomez, L., & Karatzas, D. (2020). Exploring hate speech detection in multimodal publications. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1470–1478). IEEE. https://doi.org/10.1109/WACV45572.2020.9093414

Ibrohim, M. O., & Budi, I. (2018). A dataset and preliminaries study for abusive language detection in Indonesian social media. *Procedia Computer Science, 135*, 222–229. https://doi.org/10.1016/j.procs.2018.08.169

Ibrohim, M. O., & Budi, I. (2019). Multi-label hate speech and abusive language detection in Indonesian Twitter. In *Proceedings of the Third Workshop on Abusive Language Online* (pp. 46–57). Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-3506

Ibrohim, M. O., & Budi, I. (2023). Hate speech and abusive language detection in Indonesian social media: Progress and challenges. *Heliyon, 9*(8), Article e18647. https://doi.org/10.1016/j.heliyon.2023.e18647

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets, 31*, 685–695. https://doi.org/10.1007/s12525-021-00475-2

Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Ringshia, P., & Testuggine, D. (2020). The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems, 33* (pp. 2611–2624).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems, 25*.

Prabhu, R., & Seethalakshmi, V. (2025). A comprehensive framework for multi-modal hate speech detection in social media using deep learning. *Scientific Reports, 15*, Article 13020. https://doi.org/10.1038/s41598-025-94069-z

Salim, C. E. R., & Suhartono, D. (2021). Long short-term memory for hate speech and abusive language detection on Indonesian YouTube comment section. In *2021 the 11th International Workshop on Computer Science and Engineering* (pp. 193–200). https://doi.org/10.18178/wcse.2021.06.029

Shang, L., Zhang, Y., Zha, Y., Chen, Y., Youn, C., & Wang, D. (2021). AOMD: An analogy-aware approach to offensive meme detection on social media. *Information Processing & Management, 58*(5), Article 102664. https://doi.org/10.1016/j.ipm.2021.102664

Suryawanshi, S., Chakravarthi, B. R., Arcan, M., & Buitelaar, P. (2020). Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying* (pp. 32–41). European Language Resources Association.

Thapa, S., Shah, A., Jafri, F. A., Naseem, U., & Razzak, I. (2022). A multi-modal dataset for hate speech detection on social media: Case-study of Russia-Ukraine conflict. In *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)* (pp. 1–6). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.case-1.1

Thapa, S., Jafri, F., Hürriyetoğlu, A., Vargas, F., Lee, R. K. W., & Naseem, U. (2023). Multimodal hate speech event detection—Shared task 4, CASE 2023. In *Proceedings of the 6th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text* (pp. 151–159). INCOMA Ltd.

Thapa, S., Rauniyar, K., Jafri, F., Veeramani, H., Jain, R., Jain, S., Vargas, F., Hürriyetoğlu, A., & Naseem, U. (2024). Extended multimodal hate speech event detection during Russia-Ukraine crisis—Shared task at CASE 2024. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)* (pp. 221–228). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.case-1.31

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM, 64*(3), 107–115. https://doi.org/10.1145/3446776