

## Emotion Classification in Indonesian Language: A CNN Approach with Hyperband Tuning

Muhammad Yeza Baihaqi<sup>1\*</sup>, Edmun Halawa<sup>2</sup>, Riri Asyahira Sariati Syah<sup>3</sup>, Anniza Nurrahma<sup>4</sup>, Wilbert Wijaya<sup>5</sup>

<sup>1</sup>Graduate Institute of Automation and Control, National Taiwan University of Science and Technology,

<sup>2</sup>Mechanical Engineering, President University,

<sup>3</sup>Environmental Engineering, President University,

<sup>4</sup>Psychology, Jakarta State University,

<sup>5</sup>Electrical Engineering, President University

Email: <sup>1</sup>muhammadyezab@gmail.com, <sup>2</sup>edmund.halawa@gmail.com,

<sup>3</sup>ririasyahiras@gmail.com, <sup>4</sup>anniza.nurrahma@gmail.com, <sup>5</sup>wilbertselamat@gmail.com

**Abstrak.** *Klasifikasi Emosi dalam Bahasa Indonesia: Pendekatan CNN dengan Hyperband Tuning.* Saat ini, teknik klasifikasi emosi yang andal sangat dibutuhkan di beberapa bidang. Penelitian ini mengusulkan penggunaan Convolutional Neural Network (CNN) yang telah dioptimalkan dengan Hyperband Tuner (HT) untuk secara efektif melakukan tugas Klasifikasi Emosi dalam bahasa Indonesia. Eksperimen pada berbagai teknik ekstraksi fitur, termasuk CountVectorizer (CV), TF-IDF, dan Keras Tokenizer (KT) dilakukan juga untuk mengeksplorasi kombinasi terbaik dari ekstraksi fitur dan CNN pada set data yang ada. Metodologi yang diusulkan dievaluasi dan dibandingkan dengan K-Nearest Neighbors (KNN), Decision Tree (DT), Naive Bayes (NB), dan Boosting SVM. Hasil percobaan menunjukkan bahwa metode yang digunakan pada penelitian ini mengungguli teknik yang ada, dibuktikan oleh metrik akurasi, presisi, daya ingat, dan skor F1, yang masing-masing mencapai 71,5655%, 71,5483%, 71,5655%, dan 71,0041%.

**Kata Kunci:** *convolutional neural network, ekstraksi fitur, hyperband tuner, klasifikasi emosi*

**Abstract.** *Emotion Classification in Indonesian Language: A CNN Approach with Hyperband Tuning.* In today's world, there is a high demand for accurate techniques to classify emotions in various fields. This study proposed utilizing a Convolutional Neural Network (CNN) optimized with a Hyperband Tuner (HT) to perform the Emotion Classification task in the Indonesian language effectively. Various feature extraction techniques experiments were conducted to explore the best combinations of feature extraction and CNN for the data set, including CountVectorizer (CV), TF-IDF, and Keras Tokenizer (KT). Last, the proposed methodology was evaluated and compared to the state-of-the-art techniques, including K-Nearest Neighbors (KNN), Decision Tree (DT), Naive Bayes (NB), and Boosting SVM. The experimental results revealed that the proposed method in this research outperforms the existing technique as evidenced by the accuracy, precision, recall, and F1-score metrics, which respectively reached 71.5655%, 71.5483%, 71.5655%, and 71.0041%.

**Keywords:** *convolutional neural network, emotion classification, feature extraction, hyperband tuner*

### 1. Introduction

Emotions are crucial to human intellect, encompassing making decisions rationally, interacting with others, attitude, thought, learning, and creativity [1]. Consequently, understanding and accurately detecting emotions have increased essential considerations in some fields related to decision-making, such as neuroscience, psychology, and behavioral science. Furthermore, it is vital for human-robot interaction to enable robots to provide empathetic responses to human needs, particularly in applications such as chatbots and customer service robots [2], [3].

Once emotions are detected, the subsequent critical step involves classifying them into specific emotion states, predefined classes, known as emotion classification. Despite recent

advancements, exploring emotion classification beyond four categories still needs to be improved. Yudha and Riyanarto employed Support Vector Machine (SVM), Naïve Bayes (NB), and K-Nearest Neighbors (KNN) to classify five distinct emotion types [4], revealing NB’s superior performance with 63% accuracy. Similarly, Ahmad Zamsuri *et al.* utilized KNN for classification [5], but the model’s accuracy remained below 60% for six emotion types. Moreover, model performance dropped when dealing with imbalanced datasets [6].

In recent years, the landscape of Emotion Classification has been substantially influenced by Convolutional Neural Network (CNN), a learning technique known for its ability to automatically learn and extract intricate features from raw data. However, its effectiveness relies on properly tuning their hyperparameters, a critical yet time-consuming process. Inadequate parameter tuning can underscore the significance of meticulous hyperparameter tuning [7].

In this study, our objective was to tune the hyperparameters of a CNN model for classifying emotion in the Indonesian language. A Hyperband Tuner (HT) was employed, which provided an efficient approach to tuning the hyperparameter. Additionally, an analysis was conducted to determine the optimal feature extraction methods and CNN configurations for a specific dataset. Various techniques for feature extraction, including CountVectorizer (CV), Term Frequency-Inverse Document (TF-IDF), and Keras Tokenizer (KT), were evaluated and compared with state-of-the-art techniques such as NB [4], [8], KNN [5], [9], Decision Tree (DT) [10], and Boosting SVM [11]. Combining the findings from hyperparameter optimization, feature extraction analysis, and comparative evaluation, identifying the most suitable combination techniques yielding the highest performance in the emotion classification task for the Indonesian language can be done. Furthermore, the potential applications of emotion classification in business and psychology were briefly discussed.

## 2. Literature Review

### 2.1. Feature Extraction

Feature extraction is a process of extracting essential parts of raw data. In Natural Language Processing (NLP), the data obtained through extraction consists of an essential word set. Several prevalent techniques for feature extraction from a set of words include the following:

#### 2.1.1. CountVectorizer (CV)

CountVectorizer (CV) technique is an approach for quantifying words. It counts the frequency of each term in a given text, thereby earning the moniker of raw count methodology. This technique will generate a matrix that contains mostly zero elements, commonly known as a feature matrix. To gain a deeper understanding of the CountVectorizer technique, matrix processing will be exemplified using the sample dataset provided in Figure 1.

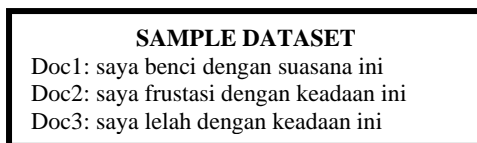


Figure 1. Sample Dataset

This method’s initial step is determining the unique word of the entire document. Based on the dataset in Figure 1, the unique words extracted are “benci”, “dengan”, “frustrasi”, “ini”, “keadaan”, “lelah”, “saya”, and “suasana”. Employing the CV, the occurrence of each unique word in every document is tabulated, as depicted in Table 1. The highlighted area represents the feature matrix of CV.

Table 1. Feature Matrix of CountVectorizer

	benci	dengan	frustrasi	ini	keadaan	lelah	saya	suasana
Doc 1	1	1	0	1	0	0	1	1
Doc 2	0	1	1	1	1	0	1	0
Doc 3	0	1	0	1	1	0	1	0

**2.1.2. Term Frequency – Inverse Document Frequency (TF-IDF)**

TF-IDF is a quantitative measure that indicates the significance of specific terms concerning particular documents. Different from CountVectorizer, which only counts the frequency of the word. The TF-IDF is calculated by multiplying the Term Frequency (TF) with the Inverse Document Frequency (IDF). The TF of a unique word in the particular document is calculated using Equation 1.

$$TF(i, j) = \frac{\text{Number of } j \text{ appear in } i}{\text{Total number of term in } i} \tag{1}$$

where,  $TF(i, j)$  denotes the term frequency of a unique word  $j$  within a particular document  $i$  in a dataset. Next, a specific word’s IDF is calculated using Equation 2.

$$IDF(j) = \log \frac{N}{DF(j)} \tag{2}$$

where,  $IDF(j)$  represents the Inverse Document Frequency of a unique word  $j$ , calculated as the logarithm of the ratio between the total number of documents ( $N$ ) in the dataset divided by the number of documents containing  $j$  ( $DF(j)$ ). Last, by multiplying Equation 1 and 2, TF-IDF output for a unique word  $j$  within a particular document  $i$  denoted by  $W(i, j)$  can be calculated using Equation 3.

$$W(i, j) = TF(i, j) \times IDF(j). \tag{3}$$

Table 2 shows the implementation of the TF-IDF technique using the dataset from Figure 1 to provide additional clarity and enhance comprehension of TF-IDF. The highlighted columns represent the resulting feature matrix obtained through TF-IDF.

**Table 2. Feature Matrix of TF-IDF**

Word ( $j$ )	TF ( $i,j$ )			DF ( $j$ )	IDF ( $j$ )	W ( $i,j$ )		
	Doc1 ( $i=1$ )	Doc2 ( $i=2$ )	Doc3 ( $i=3$ )			Doc1	Doc2	Doc3
benci	1/5	0	0	1	0.477	0.095	0.000	0.000
dengan	1/5	1/5	1/5	3	0.000	0.000	0.000	0.000
frustasi	0	1/5	0	1	0.477	0.000	0.095	0.000
ini	1/5	1/5	1/5	3	0.000	0.000	0.000	0.000
keadaan	0	1/5	1/5	2	0.176	0.000	0.035	0.035
lelah	0	0	1/5	3	0.477	0.000	0.000	0.095
saya	1/5	1/5	1/5	3	0.000	0.000	0.000	0.000
suasana	1/5	0	0	1	0.477	0.095	0.000	0.000

**2.1.3. Keras Tokenizer (KT)**

Keras Tokenizer function divides textual data into tokens the model can understand. The method works with a Keras model to preprocess textual data and then produce tokenized representations of the textual data that the model can use for further processing [12]. In contrast to CV and TF-IDF, the Keras Tokenizer requires an initial index of words before converting each document in the dataset into a numerical representation.

As an illustrative example, Figure 2 shows the list of word indices corresponding to the sample dataset depicted in Figure 1. In addition, the index number “1” is exclusively reserved to represent out-of-vocabulary (OOV) words, denoting words that have never been indexed.

KT INDEX									
OOV	: 1	dengan	: 3	ini	: 5	lelah	: 7	suasana	: 9
benci	: 2	frustasi	: 4	keadaan	: 6	saya	: 8		

**Figure 2. Keras Tokenizer Index**

To form the feature matrix, each word within a document is replaced by its corresponding index. Therefore, the “Doc 1” is turned into [8 2 3 9 5], where “saya” corresponds to 8, “benci” corresponds to 2, “dengan” corresponds to 3, “suasana” corresponds to 9, and “ini” corresponds to 5. The process as mentioned earlier is also applied to the remaining documents of the dataset. Table 3 presents the feature matrix generated by the KT technique applied to the sample dataset depicted in Figure 1.

**Table 3. Feature Matrix of KT**

Doc 1	8	2	3	9	5
Doc 2	8	4	3	6	5
Doc 3	8	7	3	6	5

## 2.2. Convolutional Neural Network (CNN)

CNN is part of deep learning technology. The utilized methodology of this technology involves the mathematical concept of convergence, as opposed to the conventional approach of matrix multiplication, in at least one of its layers. Generally, the technique was employed to identify an image by ascertaining the optimal classification for the given input [13].

CNN comprises three primary layers, namely the convolutional layer, pooling layer, and fully connected layer. Figure 3 depicts the general architecture of the CNN. The Convolutional Layer performs computations on the input data by filtering the data, while the pooling layer selects the maximum value from the filtered matrix. Finally, the fully connected layer calculates the class score. Subsequently, the resultant outcome will be categorized according to the class score, utilizing the softmax activation function for multiclass classification. Softmax activation function ( $\sigma(z_i)$ ) is shown in the Equation 4.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (4)$$

where  $e^{z_i}$  is the exponential of the input vector number  $i$ -th and sums up the exponential values of all  $n$  classes in the input vector.

CNN configuration requires variables, namely hyperparameters. Network structure hyperparameters fall into two categories. First, hyperparameters network structure includes kernel size, kernel types-value, stride, padding, hidden layer, and activation function. Second, hyperparameter network training has learning rate, momentum, number of epochs, and batch size. The value of its hyperparameters influences the performance of a CNN. Tuning errors can potentially decrease a system’s overall performance [7].

## 2.3. Hyperband Tuner (HT)

The Hyperband Tuner (HT) is a technique employed for the automated optimization of system functions. Hyperparameter tuning involves evaluating various models to identify the optimal model with the lowest time cost. The present study employs the hyperband algorithm to optimize the hyperparameter selection process. In theory, this methodology exhibits the potential to adapt to unfamiliar convergence as a parameter function. Thus, utilizing HT can resolve the intricacy of the hyperparameter amalgamation. Moreover, the performance of these tuning mechanisms exceeds that of the frequently employed Bayesian optimization technique [7].

In the HT, a predefined budget is allocated to a collection of hyperparameter configurations. Each configuration is trained and evaluated, and the algorithm discards the worst-performing half while retaining the better-performing half. This iterative process continues with progressively more resources allocated to the promising configurations. By adapting its allocation of resources, Hyperband efficiently explores a wide range of hyperparameter combinations, facilitating quicker convergence.

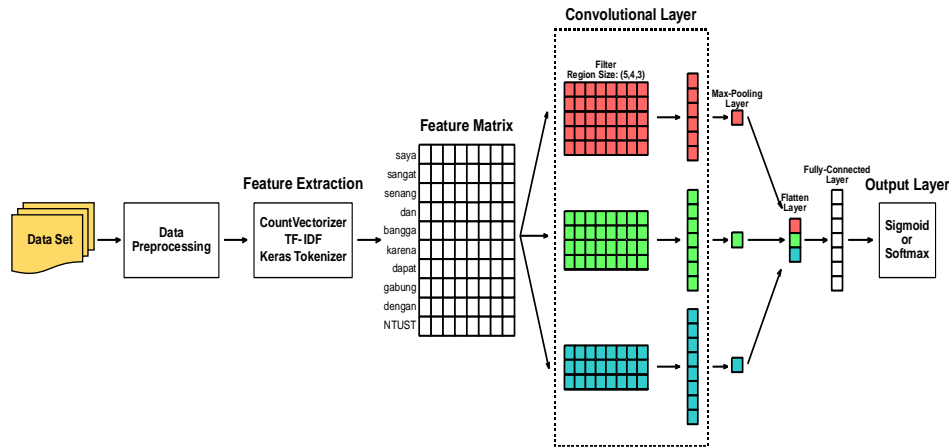


Figure 3. General Architecture of CNN

### 3. Methodology

#### 3.1. Dataset

The utilized dataset in this study was curated by Riccosan et al. [14], comprising 7,080 sentences annotated with six distinct emotion labels: anger, fear, joy, love, sadness, and neutral. Specifically, the dataset consisted of 1,130 anger-related sentences, 911 fear-related sentences, 1,275 joy-related sentences, 760 love-related sentences, 1,003 sadness-related sentences, and 2,001 neutral sentences. The dataset was partitioned into 80% training and 20% testing sets. Moreover, during the training process, 20% of the training data was reserved for validation sets.

#### 3.2. Data Preprocessing

Data preprocessing is an essential step before inputting the dataset into the model. In the initial preprocessing of this study, special characters and punctuation were removed. The dataset analysis revealed the presence of non-emotional characters such as “ϕϕ”. Subsequently, double spaces were eliminated and repeated characters were normalized. Certain irrelevant repeated characters were identified within this dataset, such as “Mahalll” which was original “Mahal”. In addition, non-emotion-related details, such as time adverbs (“malam”, “pagi”) were removed.

Furthermore, we improved the text by removing stop words using the Natural Language Toolkit (NLTK) and converting slang words based on the dictionary established by Saputri *et al.* [15]. The dataset was subsequently stemmed and lemmatized using the Sastrawi Library [16].

#### 3.3. Implementation of CNN with Hyperband Tuner

The experiment used the Google Collaboratory platform to conduct our deep learning research. Specifically, high-ram instance (32 GB) with NVIDIA V100 GPU was used. These high-ram environment and V100 GPU were crucial for obtaining reliable experimental results.

In implementing HT, validation accuracy was selected as the objective and trained the models for a maximum of 50 epochs. The factor was set to 3 and enabled overwriting of previous results. Additionally, to prevent overfitting, the stop\_early callback function was utilized. Subsequently, given the imbalanced dataset, inverse class frequency (ICF) weighting technique was applied to address the skewed distribution of the classes. ICF weighting assigns higher weights to the minority class and lower weights to the majority class. The computation of class weights for each class can be performed using Equation 5,

$$CW(c) = \frac{N_A}{(C * N_c)} \quad (5)$$

where,  $CW(c)$  represents the class weight for class  $c$ ,  $N_A$  is the total number of documents in the dataset,  $C$  is the total class in the dataset, and  $N_c$  is the total number of documents in the dataset from class  $c$ . Once the class weights for all classes in the dataset have been obtained, they were normalized using Equation 6,

$$NCW(c) = \frac{CW(c)}{TCW} \tag{6}$$

where,  $NCW(c)$  represent the normalized class weight for class  $c$  and  $TCW$  is the total class weight.

In the combination of HT and CNN with CV (CV+CNN+HT) and the combination of HT and CNN with TF-IDF (TFIDF+CNN+HT) models, the focus was on tuning the following parameters: the unit configuration and activation function of the input dense layer, the target shape of the reshape layer, the filters and kernel size of the CNN layer, the activation function of the CNN layer, the unit configuration and activation function of the second dense layer (after Global Max Pooling), and the dropout rate of the dropout layer.

In the case of the combination with KT (KT+CNN+HT), a different approach was adopted for the input layer. Instead of using a dense layer, an embedding layer was utilized. In this case, the tuned parameter was the embedding size, as opposed to the unit configuration of the dense layer. The remaining tuned parameters for KT+CNN+HT were the same as those used for CV+CNN+HT and TFIDF+CNN+HT.

The hyperband tuning process was conducted three times for each combination (CV+CNN+HT, TFIDF+CNN+HT, KT+CNN+HT). Subsequently, the model was trained using the tuned parameters five times and calculated the mean performance as a measure of stability. Ensuring a stable model was crucial, as a significant performance gap indicated suboptimal parameter selection. Thus, the highest achieved performance in each category was compared and examined as the best-tuned parameters from each combination. As a result, the best-tuned parameters from the optimal combination were compared with previous research methods to assess their effectiveness and relevance. Figure 4 shows the implementation of CNN+HT with several feature extractions.

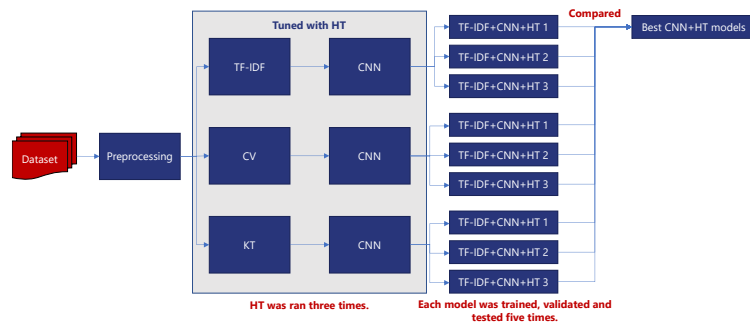


Figure 4. Implementation of CNN+HT with Several Feature Extractions

### 3.4. Metrics

In this research, four metrics were used to evaluate the model performance. Those metrics are accuracy, precision, recall, and F1-score. The equation of accuracy, precision, recall, and F1-score are presented respectively in Equation 7, 8, 9, and 10.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{10}$$

where *TP* (True Positive) represents the instances accurately classified into a particular class, while *TN* (True Negative) refers to the instances correctly classified as not belonging to that class. *FP* (False Positive) indicates the instances that have been incorrectly classified to a specific class. Conversely, *FN* (False Negative) refers to the instances that have been incorrectly classified.

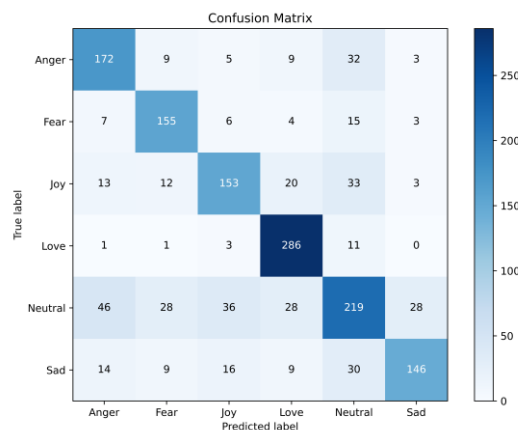
## 4. Result and Discussion

### 4.1. Feature Extraction+CNN+HT

The results of applying HT to CNN with distinct feature extraction techniques are presented in Table 4. The findings demonstrate high consistency in the tuning outcomes across the three feature extraction methods, with a marginal deviation of approximately 1%. Among the three methods investigated, it has been observed that utilizing KT as a feature extraction approach yields superior results compared to the other two techniques. The KT+CNN method consistently achieves performance levels of over 71% across multiple evaluation metrics, including accuracy, precision, recall, and F1-score. This performance surpassed that of the CNN model with alternative feature extraction methods ranging from 67% to 68%, demonstrating its overall superiority. The best-tuned KT+CNN+HT model (KT+CNN+HT 3) identified in this study achieved an accuracy of 71.5655%, precision of 71.5483%, recall of 71.5655%, and F1-score of 71.0041%. In addition, the confusion matrix of the KT+CNN+HT 3 is shown in Figure 5.

**Table 4. Performance Comparison of CNN+HT with Several Feature Extractions**

CNN Model	Tuned Parameter	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
TF-IDF+CNN+HT	1	68.1534 ± 0.63	68.8826 ± 0.36	68.1534 ± 0.63	68.0580 ± 0.57
	2	67.8978 ± 0.52	68.2376 ± 0.68	67.8978 ± 0.52	67.7607 ± 0.47
	3	67.7188 ± 0.65	68.6158 ± 0.63	67.7188 ± 0.73	67.6750 ± 0.75
CV+CNN+HT	1	67.5783 ± 0.56	68.3793 ± 1.00	67.5783 ± 0.56	67.5547 ± 0.48
	2	68.0511 ± 0.33	68.7359 ± 0.75	68.0511 ± 0.33	67.9747 ± 0.37
	3	67.5399 ± 0.43	68.0759 ± 0.43	67.5399 ± 0.43	67.4203 ± 0.40
KT+CNN+HT	1	70.4665 ± 0.52	70.3610 ± 0.54	70.4537 ± 0.52	69.6692 ± 0.63
	2	71.0160 ± 0.95	70.9931 ± 0.81	71.0160 ± 0.95	70.5472 ± 1.006
	3	71.5655 ± 0.49	71.5483 ± 0.44	71.5655 ± 0.49	71.0041 ± 0.57



**Figure 5. Confusion Matrix of KT+CNN+HT 3**

### 4.2. Discussion and Comparison

Based on our experiment, the HT showed great performance by consistently generating stable CNN models comparable to diverse feature extraction methods, including TF-IDF, CV, and KT. From our experience, KT offers notable advantages over CV and TF-IDF as the feature



extraction in CNN architecture. KT enables the direct feeding of tokenized sequences, obviating the need for data reshaping. Additionally, KT provides automatic padding and truncation, streamlining the input preprocessing stage.

The outcomes derived from our experimental analysis unequivocally establish the superiority of KT relative to alternative feature extractions. KT's exceptional performance can be attributed to its effective handling of out-of-vocabulary (OOV) words. It is highly possible that the model encountered words in the test set but not in the training set. KT provided a mechanism to handle this OOV issue by assigning a specific token to represent it. Another contributing factor to the enhanced performance of KT compared to TF-IDF and CV was its natural ability to provide word-level representation. By providing word-level representation, KT could capture more specific meaning, context, and dependencies between words compared to TF-IDF and CV.

While KT demonstrated superior performance in our experiments, it also presented a drawback. It demanded a more extended training process than CV and TF-IDF techniques. The study revealed that KT's average duration per epoch was 5.125 seconds, significantly longer than the 3 seconds for TF-IDF and the quickest 2.25 seconds per epoch observed with CV. These disparities underscored the heightened computational overhead of employing KT.

Furthermore, a comparative analysis was conducted to assess the performance of our tuned CNN model (KT+CNN+HT) against other state-of-the-art techniques. While it was observed that the precision of our model was slightly lower than that of TFIDF combined with Boosting SVM [11]. The comprehensive evaluation in Table 6 indicated that our tuned CNN model outperformed previous techniques regarding accuracy, recall, and F1 score.

**Table 6. Comparison with the State-of-Art Techniques**

No	Technique	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1	TF-IDF+MNB [4]	66.8300	69.1200	67.0400	67.2100
2	TF-IDF+KNN [5]	62.8700	62.5000	63.5200	62.6000
3	TF-IDF+CNB [8]	67.6760	68.8724	68.3130	67.6955
4	CV+KNN [9]	67.2800	67.5400	67.5200	66.9800
5	CV+DT [10]	70.2800	70.6700	70.6000	70.3300
6	TF-IDF+Boosting SVM [11]	62.1725	73.7281	58.3936	61.4729
7	KT+CNN+HT (ours)	71.5655	71.5483	71.5655	71.0041

However, despite surpassing the performance of previous techniques, our model performance of 71.5655% accuracy in text-based Indonesian emotion classification underscores the intricate challenges involved. The scarcity of linguistic resources due to the low-resource nature of the Indonesian language hampers our model's ability to capture nuanced language intricacies. Despite efforts to mitigate class imbalance in the dataset, accurate prediction of minority classes remains hindered. The complexity of multiclass classification, encompassing six distinct emotions, further exacerbates the challenge, straining predictive accuracy. Additionally, the utilized dataset covers a wide array of topics, introducing a further layer of complexity that is generally more challenging than a single-topic dataset. The presence of lengthy sentences within the dataset exacerbates this issue by introducing noise and reducing precision. Nevertheless, this advancement holds significant value by exceeding the outcomes of preceding methods. It forms an essential foundation for future work, displaying promise for real-world applications.

#### **4.3. The Potential of Emotion Classification in Psychology**

The application of emotion classification holds considerable potential within the realm of psychology. Emotion classification is a valuable tool for various psychological applications, including predicting loneliness through analysis of social media accounts [17]. Furthermore, it supports foundational research in psychology, facilitating investigations into fundamental questions such as the factors that contribute to human thriving, the determinants of happiness, and how individuals express emotions through mining social media posts [18].

However, although emotion classification models demonstrate the ability to accurately classify text inputs by leveraging inherent patterns and features within the text, it is imperative to acknowledge that textual inputs may not consistently reflect the actual emotional state of



individuals [18]. Factors such as the deliberate downplaying or exaggeration of emotions, the utilization of euphemisms, and the employment of other linguistic strategies can misalign with genuine feelings. Furthermore, it is worth noting that most existing emotion classification datasets encompass only six basic emotions, despite numerous emotional states. Therefore, in current development, emotion classification should be regarded as a reference or tool for psychologists and other professionals when making decisions or assessments related to emotional states rather than providing a comprehensive understanding of an individual's emotions [19].

#### 4.4. The Potential of Emotion Classification in Business

Emotion classification within a business context presents considerable advantages to companies. The primary function of this tool is to aid in the acquisition of significant perspectives on consumer opinion, thereby promoting a comprehensive comprehension of customers' emotional reactions toward their products, services, or brand. This knowledge enables companies to make well-informed decisions, and efficiently tackle customer issues. Furthermore, precise classification of customer emotions allows organizations to customize their interactions and promotional strategies, increasing customer involvement and allegiance [20]. However, preceding the implementation of emotion classification, it is crucial to undertake meticulous deliberations such as system integration and scalability, data privacy and ethical practices, and cost.

### 5. Conclusions and Recommendations

The findings indicate that employing the CNN methodology with HT yields noteworthy outcomes. The best CNN model combinations were achieved by combining CNN with KT tuned by HT (KT+CNN+HT). While the precision values exhibited by KT+CNN+HT are 2% inferior to those of the TFIDF+Boosting SVM approach, notable disparities in outcomes are evident across other metrics. The combination results in an accuracy score of 71.5655%, precision score of 71.5483%, recall score of 71.5655%, and F1-score of 71.0041%. Lastly, forthcoming research may involve crafting ensemble models fine-tuned through Hyperparameter Tuning (HT) while integrating cutting-edge techniques tailored to address imbalanced datasets.

Furthermore, the prospective applications of emotion classification within the domains of business and psychology were briefly examined. The categorization of emotions holds promise as a guiding framework for psychologists in formulating decisions about emotional states within psychology's purview. Analogously, within the context of business, the integration of this technological capability stands to augment entrepreneurs' capacity to discern the extent of consumer demand for specific products or services, thereby engendering a more versatile approach. The latent possibilities within these domains warrant dedicated exploration and analysis in forthcoming investigations.

### References

- [1] Y. Du, R. G. Crespo, and O. S. Martínez, "Human emotion recognition for enhanced performance evaluation in e-learning," *Progress in Artificial Intelligence*, vol. 12, no. 2, pp. 199–211, June 2023.
- [2] W. Graterol, J. Diaz-Amado, Y. Cardinale, I. Dongo, E. Lopes-Silva, and C. Santos-Libarino, "Emotion Detection for Social Robots Based on NLP Transformers and an Emotion Ontology," *Sensors*, vol. 21, no. 4, pp. 1–19, February 2021.
- [3] M. Y. Baihaqi and S. S.-D. Xu, "Seven stages of action for the interaction design of the customer service of a robot in a shopping mall," in *Proc. International Conference on Advanced Robotics and Intelligent Systems*, Taipei, Taiwan, August 24–27, 2022, p.1.
- [4] B. Y. Pratama and R. Sarno, "Personality classification based on Twitter text using Naive Bayes, KNN and SVM," in *Proc. International Conference on Data and Software Engineering*, Yogyakarta, Indonesia, November 25–26, 2015, pp. 170–174.
- [5] A. Zamsuri, S. Defit, and G. W. Nurcahyo, "Classification of multiple emotions in Indonesian text using the k-nearest neighbor method," *Journal of Applied Engineering and Technological Science*, vol. 4, no. 2, pp. 1012–1021, June 2023.

- [6] E. Nugraheni, "Indonesian twitter data pre-processing for the emotion recognition," in *Proc. International Seminar on Research of Information Technology and Intelligent Systems*, Yogyakarta, Indonesia, December 5–6, 2019, pp. 58–63.
- [7] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, April 2018.
- [8] N. Hayatin, S. Alias, L. Hung, and M. Sainin, "Sentiment analysis based on probabilistic classifier techniques in various Indonesian review data," *Jordanian Journal of Computers and Information Technology*, vol. 8, no. 3, pp. 271–282, September 2022.
- [9] B. Harjito, A. Wijayanto, K. N. Aini, and B. Murtiyas, "Comparison of multinomial naïve bayes with k-nearest neighbors, support vector machine and random forest for classification of 'network attacks' document," in *Proc. International Conference on Informatics and Computing*, Manado, Indonesia, October 16–17, 2019, pp. 1–6.
- [10] A. Rusli, A. Suryadibrata, S. B. Nusantara, and J. C. Young, "A comparison of traditional machine learning approaches for supervised feedback classification in Bahasa Indonesia," *International Journal of New Media Technology*, vol. 7, no. 1, pp. 28–32, July 2020.
- [11] A. Maulana and H. Pratiwi, "Sentiment analysis of public towards infrastructure development in Indonesia on Twitter media using boosting support vector machine method," in *Proc. International Conference on Science and Applied Science*, Jakarta, Indonesia, July 20, 2019, pp. 020082-1–020082-13.
- [12] M. V. S. Rishita, M. A. Raju, and T. A. Harris, "Machine translation using natural language processing," *MATEC Web of Conferences*, vol. 277, no. 02004, pp. 1–6, April 2019.
- [13] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, no. 23, pp. 1–31, May 2017.
- [14] Riccosan, K. E. Saputra, G. D. Pratama, and A. Chowanda, "Emotion dataset from Indonesian public opinion," *Data Brief*, vol. 43, p. 108465, Aug. 2022, doi: 10.1016/j.dib.2022.108465.
- [15] M. S. Saputri, R. Mahendra, and M. Adriani, "Emotion classification on Indonesian twitter dataset," in *Proc. International Conference on Asian Language Processing*, Bandung, Indonesia, November 15–18, 2018, pp. 90–95.
- [16] A. Librian, "Sastrawi: Stemmer Bahasa Indonesia (Indonesian Stemmer)," 2016. [Online]. Available: <https://github.com/sastrawi/sastrawi>. [Accessed: 25-May-2023].
- [17] S. Kiritchenko, W. Hipson, R. Coplan, and S. M. Mohammad, "SOLO: A corpus of tweets for examining the state of being alone," in *Proc. Language Resources and Evaluation Conference*, Marseille, France, June 20–25 2020, pp. 1567–1577.
- [18] S. M. Mohammad, "Ethics Sheet for Automatic Emotion Recognition and Sentiment Analysis," *Computational Linguistics*, vol. 48, no. 2, pp. 239–278, June 2022.
- [19] J. C. Jackson, J. Watts, J.-M. List, C. Puryear, R. Drabble, and K. A. Lindquist, "From text to thought: How analyzing language can advance psychological science," *Perspectives on Psychological Science*, vol. 17, no. 3, pp. 805–826, May 2022.
- [20] A. Haleem, M. Javaid, M. Asim Qadri, R. Pratap Singh, and R. Suman, "Artificial intelligence (AI) applications for marketing: A literature-based study," *International Journal of Intelligent Networks*, vol. 3, pp. 119–132, September 2022.