Development of a Littering Behavior Detection Using 3D Convolutional Neural Networks (3D CNN)

Nyayu Latifah Husni[1], Ekawati Prihatini[2], Monica Ulandari[3], Ade Silvia Handayani

Department of Electrical Engineering, Faculty of Electrical Engineering, Politeknik Negeri sriwijaya
nyayu_latifahl@polsri.ac.id

ABSTRACT

Littering has become a significant problem that negatively impacts public health and environmental cleanliness. This research introduces an innovative solution using 3D Convolutional Neural Networks (3D CNN) technology to automatically detect littering behavior through real-time CCTV recordings. Two models were developed and tested. Model 1, which employs Conv3D, Batch Normalization, and Dropout, showed high training accuracy but exhibited fluctuations in validation accuracy, indicating potential overfitting. In contrast, Model 2, designed with a simpler structure without Batch Normalization and Dropout, achieved higher classification accuracy and efficiency. Both models significantly contribute to addressing littering in public areas, increasing awareness, and supporting environmental law enforcement. The integration of 3D CNN technology in detecting littering behavior demonstrates its potential to reduce pollution and promote environmentally responsible behavior.

Keywords: 3D Convolutional Neural Network (CNN), Littering, Environmental cleanliness, Automatic detection, Classification accuracy.

1. INTRODUCTION

Littering has become a serious issue that negatively impacts public health and environmental cleanliness. Environmental pollution caused by unmanaged waste disposal behavior has become an urgent global concern [1]-[3]. The act of littering itself can be described through body movements or the position of body parts in the context of time and gravity [4]. Analyzing human behavior and human-computer interaction has become increasingly important in understanding and addressing this issue [5]-[6].

To tackle this challenge, image-based surveillance systems are used to monitor human activities involving littering in real-time from remote locations [7]. This eliminates the need for on-site personnel to observe these activities, allowing urban authorities to address the problem more efficiently. Developing a device that can detect human behavior associated with littering offers a practical solution for remote monitoring via online platforms, reducing manual surveillance efforts [8]-[9].

This research discusses the development of a system aimed at identifying littering behavior using real-time CCTV monitoring and image processing technology, particularly Convolutional Neural Networks (CNN). The use of 3D CNN in video or image processing allows for the extraction of both spatial and temporal information, which is essential in understanding and detecting littering behavior [10].

By leveraging CCTV to record videos of environments where littering behavior is targeted for detection, this research contributes to developing innovative solutions for environmental pollution issues. Through efficient monitoring and surveillance using this technology, it is expected to raise public awareness and encourage more responsible waste disposal behavior [11].

Previous studies have shown that image-based surveillance can detect various types of human behavior in different contexts. In 2018, research by Zhang et al. demonstrated the effectiveness of using deep learning to detect suspicious behavior in video surveillance systems [12]. Additionally, in 2020, Wang et al. outlined the use of 3D CNN for video analysis in detecting human physical activities [13]. In a more specific context, in 2019, Li et al. highlighted the potential of image processing technology for environmental monitoring [14]. However, research specifically focused on developing and implementing littering behavior detection systems using 3D CNN remains limited. This study offers an original contribution by integrating advanced image processing and video surveillance technologies to address this specific environmental pollution problem [15].

2. MATERIAL AND METHOD

2.1 DATASET COLLECTION

Dataset Retrieval was taken using a mobile phone camera. Dataset collection is an important step to ensure the accuracy and effectiveness of the 3D Convolutional Neural Network (3D CNN) model in detecting trash throwing behavior. In this research, the dataset consists of 102 data labelled "WASTE_WASTE" and 96 data labelled "NORMAL", which was then converted into 11,424 data frames. To prepare this data, the dataset is divided into two main parts, namely training data and validation data. each 80% for training and 20% for validation. This division aims to ensure that the model can be trained effectively and validated to test the model's accuracy and generalization on never-before-seen data.

2.2 3D CNN ALGORITHM

In recent years, 3D convolution layer-based neural network architectures have become very popular in video data classification due to their ability to analyze the position of objects in the context of time. Additionally, 3D CNN generates 3D activation maps during the convolution step, which is important for data analysis as well as temporal and volumetric context. To compute the element representation at a low level, a three-dimensional filter is used to convolve in three dimensions.

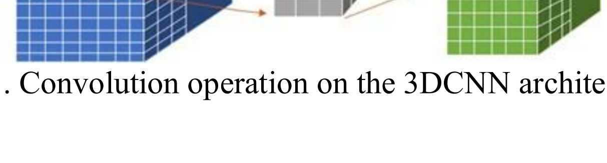

FIGURE 1. Convolution operation on the 3DCNN architecture [23].

A form of nonlinear down sampling of the input tensor is a three-dimensional MaxPooling3D layer. This method divides the input tensor data into three-dimensional subtensors along three dimensions and selects the element with the maximum numerical value from each subtensor. Finally, this method converts the input tensor into an output tensor by replacing the maximum element of each subtensor. MaxPooling3D is usually used for colour images, as shown in Figure 2 [22]-[23].

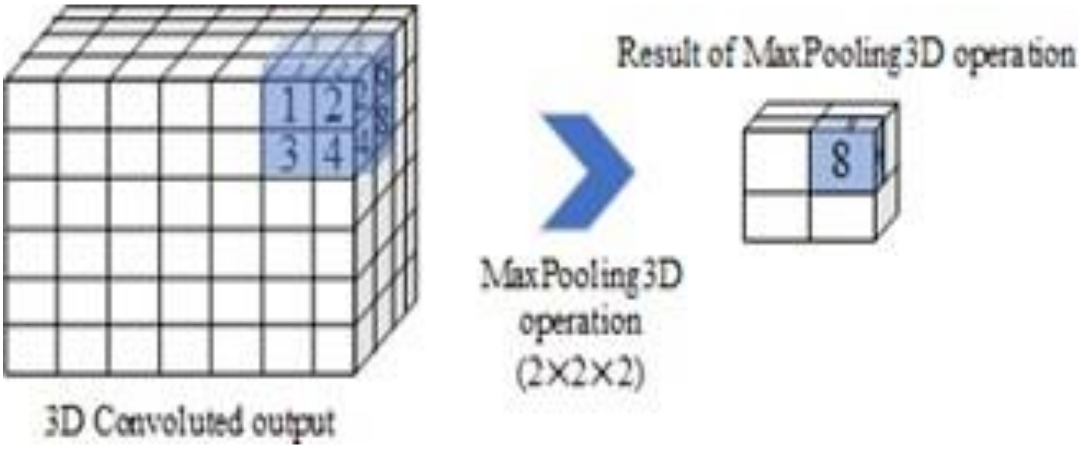

FIGURE 2. MaxPooling 3D operation in the 3DCNN architecture [23].

The clarification modelling process with the 3D CNN algorithm is carried out using the Python programming language. This process is an important part of the device that functions to detect human activity in littering because it is a machine learning stage that teaches how to use objects correctly. The initial step in the 3D CNN method is to collect a data set containing the input video. Then, the data is broken down into images and divided into training and testing data. The pre-processing process is used in data collection to produce training which is carried out simultaneously with data testing to obtain a 3D CNN model. If the model obtained is not optimal, it will enter hyper parameters and then enter the dataset stage, but if it is optimal, the model will be saved.

3. RESULTS AND DISCUSSION

3.1 TRAINING MODEL 1

Model 1 is a three-dimensional convolutional neural network (3D CNN) architecture developed specifically for processing video data with input resolution (24, 56, 32, 1). This architecture consists of several layers aimed at extracting complex spatial and temporal features. The first layer uses Conv3D with 128

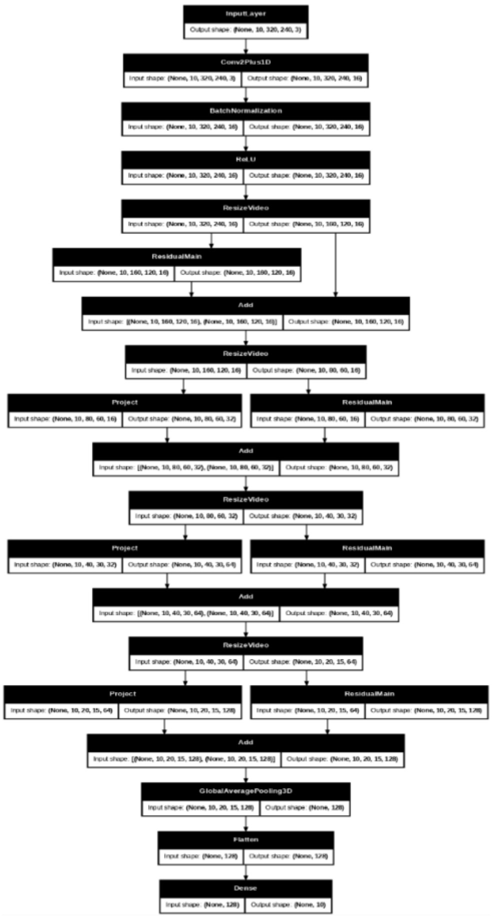
InputLayer
Output shape: (None, 10, 320, 240, 3)
Conv2Plus1D
Input shape: (None, 10, 320, 240, 3)
Output shape: (None, 10, 320, 240, 16)
BatchNormalization
Input shape: (None, 10, 320, 240, 16)
Output shape: (None, 10, 320, 240, 16)
ReLU
Input shape: (None, 10, 320, 240, 16)
Output shape: (None, 10, 320, 240, 16)
ResizeVideo
Input shape: (None, 10, 320, 240, 16)
Output shape: (None, 10, 160, 120, 16)
ResidualMain
Input shape: (None, 10, 160, 120, 16)
Output shape: (None, 10, 160, 120, 16)
Add
Input shape: [(None, 10, 160, 120, 16), (None, 10, 160, 120, 16)]
Output shape: (None, 10, 160, 120, 16)
ResizeVideo
Input shape: (None, 10, 160, 120, 16)
Output shape: (None, 10, 80, 60, 16)
Project
Input shape: (None, 10, 80, 60, 16)
Output shape: (None, 10, 80, 60, 32)
ResidualMain
Input shape: (None, 10, 80, 60, 16)
Output shape: (None, 10, 80, 60, 32)
Add
Input shape: [(None, 10, 80, 60, 32), (None, 10, 80, 60, 32)]
Output shape: (None, 10, 80, 60, 32)
ResizeVideo
Input shape: (None, 10, 80, 60, 32)
Output shape: (None, 10, 40, 30, 32)
Project
Input shape: (None, 10, 40, 30, 32)
Output shape: (None, 10, 40, 30, 64)
ResidualMain
Input shape: (None, 10, 40, 30, 32)
Output shape: (None, 10, 40, 30, 64)
Add
Input shape: [(None, 10, 40, 30, 64), (None, 10, 40, 30, 64)]
Output shape: (None, 10, 40, 30, 64)
ResizeVideo
Input shape: (None, 10, 40, 30, 64)
Output shape: (None, 10, 20, 15, 64)
Project
Input shape: (None, 10, 20, 15, 64)
Output shape: (None, 10, 20, 15, 128)
ResidualMain
Input shape: (None, 10, 20, 15, 64)
Output shape: (None, 10, 20, 15, 128)
Add
Input shape: [(None, 10, 20, 15, 128), (None, 10, 20, 15, 128)]
Output shape: (None, 10, 20, 15, 128)
GlobalAveragePooling3D
Input shape: (None, 10, 20, 15, 128)
Output shape: (None, 128)
Flatten
Input shape: (None, 128)
Output shape: (None, 128)
Dense
Input shape: (None, 128)
Output shape: (None, 10)

FIGURE 3. 3D CNN Model 1 architecture

3.2 MODEL 1 TRAINING RESULTS

Model 1 is based on training and validation data for 50 epochs. This model shows variations in the level of accuracy and loss measured on both types of data. Initially, the model starts with an accuracy of around 51.52% and gradually increases until it reaches a peak of around 96.97% during the training process. However, accuracy on validation data tends to be stable in the range between 51.52% to 56.57%, which indicates the potential for overfitting or complexity mismatch between the data used for training and validation.

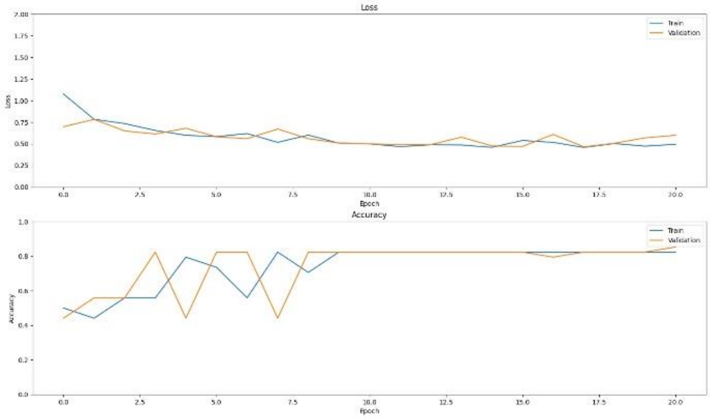

FIGURE 4. Loss and accuracy of Model 1

The first epoch started with a loss of 1.2364 and an accuracy of 51.52%, while validation metrics showed a loss of 0.6968 with an accuracy of 53.03%. As training progresses, both loss and accuracy fluctuate, indicating that the model is attempting to generalize patterns learned from the training data to previously unseen validation data. However, the main challenge faced is in consistently improving validation accuracy.

Fluctuations in validation accuracy suggest the possibility of improving regularization techniques or adjustments to the model architecture to improve the model's general ability to capture more complex patterns from video data. These findings provide an important foundation for discussing model performance in depth as well as identifying future research directions that can optimize training stability and validation accuracy in the context of similar 3D CNN applications.

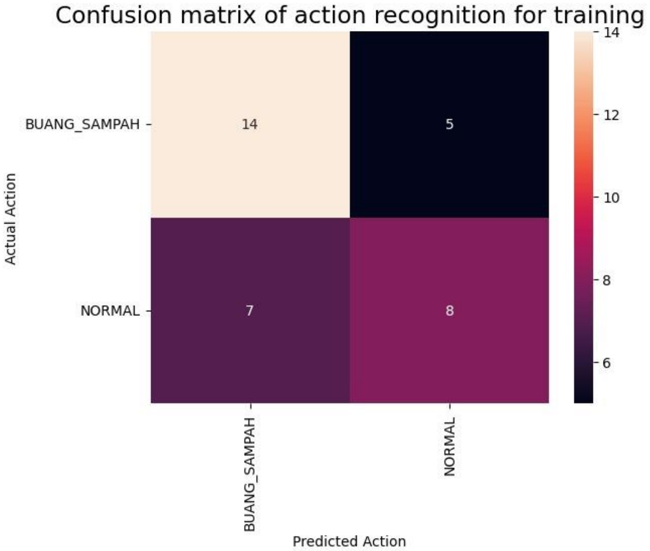

FIGURE 5. Confusion Matrix Model 1

The image above is a confusion matrix used to evaluate the model's performance in recognizing actions in training data. This matrix depicts the number of correct and incorrect predictions made by the model for two classes: "WASTE_WASTE" and "NORMAL".

From the matrix, we can see that the model succeeded in classifying 14 instances of "WASTE_WASTE" correctly, but incorrectly classified 5 instances of "WASTE_WASTE" as "NORMAL". For the "NORMAL" class, the model managed to correctly classify 8 instances, but incorrectly classified 7 "NORMAL" instances as "WASTE_WASTE".

Overall, this confusion matrix shows that the model has quite good performance in classifying the "WASTE_TRASH" action with fewer errors compared to the "NORMAL" class. However, there were some errors in the predictions of both classes, indicating areas that need to be improved to improve the overall accuracy of the model. Further analysis and hyperparameter tuning may be necessary to reduce the number of classification errors and improve the model's performance in recognizing actions from videos.

$$True\ Positive\ (TP)=14$$

True Negative (TN)=8
False Positive (FP)=7
False Negative (FN)=5

Accuracy

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{14+8}{14+8+7+5} = 0.647 = 64.7\%$$

Precision

$$Precision = \frac{TP}{TP+FP} = \frac{14}{14+7} = 0.667 = 66.7\%$$

Recall

$$Recall = \frac{TP}{TP+Fn} = \frac{14}{14+5} = 0.737 = 73.7\%$$

F1 Score

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0.667 \times 0.737}{0.667+0.737} = 0.7 = 70.0\%$$

Specificity

$$Specificity = \frac{TN}{TN+FP} = \frac{8}{8+7} = \frac{8}{15} = 0.533 = 53.3\%$$

Model 2 is a three-dimensional convolutional neural network (3D CNN) architecture designed to process video data with input resolutions (24, 56, 32, 1). This model has a simpler structure than Model 1. This architecture consists of one Conv3D layer with 32 filters and a 3x3x3 kernel, which uses a ReLU activation function to extract features from the input video. After the convolution process, the data passes through a MaxPooling3D layer with a pool size of 2x2x2 to reduce spatial dimensions and capture the most significant features of the video.

In contrast to Model 1, Model 2 does not implement BatchNormalization or Dropout, making the structure simpler and the training process faster. However, these shortcomings of BatchNormalization and Dropout also mean that Model 2 may be more susceptible to overfitting, as there is no internal mechanism to stabilize and prevent the model from learning the training data too specifically. After the pooling layer, the data is flattened into a one-dimensional vector using the Flatten layer, so that it can be processed by fully connected layers.

The resulting flatten vector is then passed to the Dense layer with 32 Units and the ReLU activation function, which is tasked with learning a higher-level feature representation from the input video. Finally, the output layer consists of Dense with 2 Units and a Softmax activation function, which generates probabilities for the two desired output classes. Although simpler, the structure of Model 2 allows for faster training and requires fewer computing resources. However, its ability to capture complex features from video data may not be as strong as Model 1 which has a more complex structure and applies regularization techniques such as BatchNormalization and Dropout.

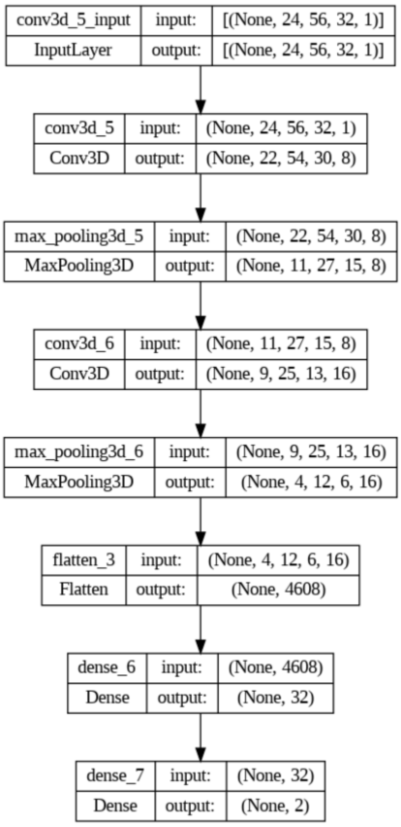
conv3d_5_input | input: | [(None, 24, 56, 32, 1)]
InputLayer | output: | [(None, 24, 56, 32, 1)]
conv3d_5 | input: | (None, 24, 56, 32, 1)
Conv3D | output: | (None, 22, 54, 30, 8)
max_pooling3d_5 | input: | (None, 22, 54, 30, 8)
MaxPooling3D | output: | (None, 11, 27, 15, 8)
conv3d_6 | input: | (None, 11, 27, 15, 8)
Conv3D | output: | (None, 9, 25, 13, 16)
max_pooling3d_6 | input: | (None, 9, 25, 13, 16)
MaxPooling3D | output: | (None, 4, 12, 6, 16)
flatten_3 | input: | (None, 4, 12, 6, 16)
Flatten | output: | (None, 4608)
dense_6 | input: | (None, 4608)
Dense | output: | (None, 32)
dense_7 | input: | (None, 32)
Dense | output: | (None, 2)

FIGURE 6. 3D CNN Model 2 architecture

The results from training the 2 3D CNN models show that performance improves with increasing epochs. During 15 epochs of model training, there was a significant increase in performance from start to finish. At Epoch 1, the model showed training accuracy of 53.37% with a loss of 2.7855, while validation accuracy was 58.33% with a loss of 0.6898. Even though it experienced fluctuations at the beginning of training, as seen in Epoch 2 with validation accuracy dropping to 41.67%, the model continued to experience consistent improvements in the following epochs.

In Epoch 3, validation accuracy increased to 68.33% with a validation loss of 0.5894. This indicates that the model is starting to learn better. The improvement continued until Epoch 6, where validation accuracy reached 80% with a validation loss of 0.4469.

Towards the last epochs, the model increasingly shows stable and accurate performance. At Epoch 10, the model achieved validation accuracy of 86.67% with a validation loss of 0.4155. Accuracy continued to increase until it reached 90% at Epoch 15, although validation loss increased slightly to 0.4861.

Overall, the results of this training show that the model succeeded in significantly improving classification accuracy with lower loss, demonstrating the model's ability to learn and adapt well to the given data.

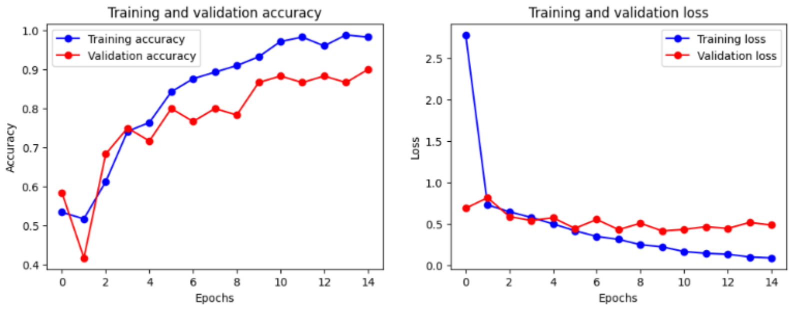

FIGURE 7. Loss and Accuracy Model 2

The image above displays two graphs that compare accuracy and loss in the model training and validation process for 14 epochs. The graph on the left shows the training and validation accuracy, while the graph on the right shows the training and validation loss.

In the accuracy graph (left), it can be seen that training accuracy (blue line) increases consistently from around 53% in the first epoch to around 98% in the 14th epoch. Validation accuracy (red line) also increases, but with fluctuations, from around 58% in the first epoch to around 90% in the 14th epoch. This graph shows that the model succeeded in learning from the training data well and was able to generalize to the validation data with a significant increase in accuracy.

The loss graph (right) shows that the training loss (blue line) decreased sharply from around 2.8 in the first epoch to around 0.1 in the 14th epoch. Validation loss (red line) also shows a decrease, although with fluctuations, from around 0.7 in the first epoch to around 0.5 in the 14th epoch. This decrease in loss indicates that the model is getting better at minimizing prediction errors during the training process.

Overall, these two graphs show that the model experienced significant performance improvements in both accuracy and loss over the 14 epochs of training. The model demonstrated good ability to learn and generalize from training data to validation data, although there were fluctuations in validation values that may have been caused by data variations.

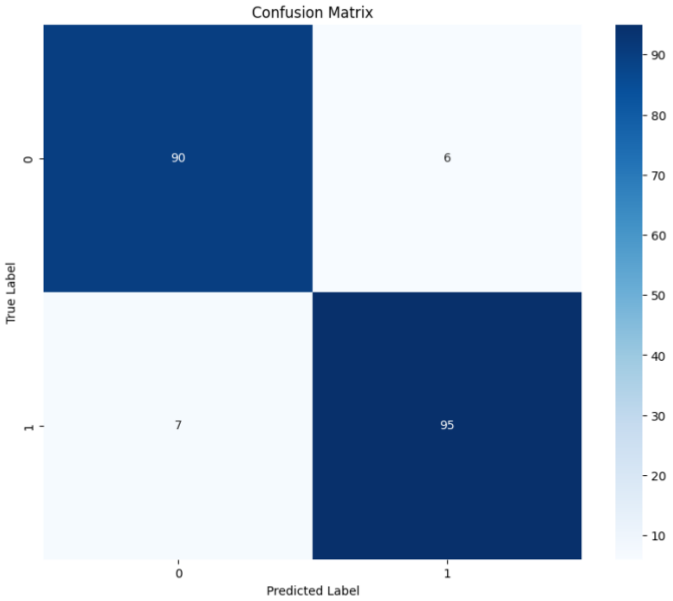

FIGURE 8. Confusion Matrix Model 2.

Based on the Confusion matrix which shows the results of model predictions on test data, it can be concluded that this model is effective in distinguishing between the two classes tested. From this matrix, it is known that correct predictions for class 0 reached 90 cases, while correct predictions for class 1 reached 95 cases. Meanwhile, prediction errors occurred in lower numbers, with 6 class 0 cases incorrectly predicted as class 1, and 7 class 1 cases incorrectly predicted as class 0. Thus, the accuracy percentage for class 0 was around 93.75%, while for class 1 it was around 93.14%. The overall total accuracy of the model is approximately 93.45%, indicating high reliability of the model in recognizing both classes. These results indicate that the model can be trusted in practical applications to detect differences between class 0 and class 1 with satisfactory accuracy.

$$\textit{True Positive (TP)}=102$$
$$\textit{True Negative (TN)}=96$$
$$\textit{False Positive (FP)}=7$$
$$\textit{False Negative (FN)}=6$$

Accuracy

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{102+96}{102+96+7+5} = 0.942 = 94.2\%$$

Precision

$$Precision = \frac{TP}{TP+FP} = \frac{102}{102+7} = 0.935 = 93.5\%$$

Recall

$$Recall = \frac{TP}{TP+Fn} = \frac{102}{102+5} = 0.953 = 95.3\%$$

F1 Score

$$\textit{F1 Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2\times 93.5\times 95.3}{93.5+95.3} = 94{,}3 = 94{,}3\%$$

Specificity

$$Specificity = \frac{TN}{TN+FP} = \frac{96}{96+7} = 0.932 = 93.2\%$$

4. CONCLUSION

The conclusions of the two drilled 3D CNN models show an interesting comparison in the context of video processing for detecting waste disposal behavior. Model 1, although complex with a structure consisting of multiple layers of convolution, Dropout, and BatchNormalization, shows high accuracy gains during training, reaching a peak accuracy of around 96.97%. However, the main challenge lies in stable accuracy on validation data, which indicates the potential for overfitting or difficulty in generalizing the model to unseen data.

Meanwhile, Model 2, despite having a simpler structure with only one convolution layer and no BatchNormalization or Dropout, shows a significant increase in accuracy over 15 epochs of training. Despite appearing early on accuracy validation, the model was able to achieve fairly high accuracy with low loss, demonstrating the ability to learn well from the given data.

Overall, both models show potential in classifying waste disposal behavior from video data with satisfactory accuracy. Model 1 provides high acoustics with a complex structure, while Model 2 shows the ability for faster training through a simpler structure. Further analysis and adjustments may be needed to improve the stability and generalizability of both in practical applications for detecting and preventing inappropriate waste disposal behavior.

ACKNOWLEDGEMENTS

Acknowledgement is extended to the Department of Electrical Engineering, Faculty of Electrical Engineering, Sriwijaya State Polytechnic, for its support and resources during this research.

REFERENCES

[1] D. Aishwarya and R. I. Minu, “Edge computing based surveillance framework for real time activity recognition,” *ICT Express*, vol. 7, no. 2,pp. 182–186, 2021, doi: 10.1016/j.icte.2021.04.010.

[2] Wibowo I. Pola perilaku kebersihan: Studi psikologi lingkungan tentang penanggulangan sampah perkotaan. Jurnal Makara, Sosial Humaniora. 2009 Jul;13(1):37-47.

[3] A. Abdelrahman, M. El-Sayed, and A. El-Sawy, "Deep learning-based approach for detection and classification of littering behavior from real-time surveillance videos," *Applied Intelligence*, pp. 1-18, 2021.

[4] S. Ahmad, M. I. Ghani, and M. A. Majid, "Convolutional Neural Network (CNN) for Image Classification of Litter in Public Places," *International Journal of Advanced Science and Technology*, vol. 29, no. 2, pp. 502-509, 2020.

[5] S. Al-Saadi, M. Al-Sarhan, and S. Al-Saadawi, "A Novel Deep Learning-Based Approach for Litter Detection and Classification Using Convolutional Neural Networks," *IEEE Access*, vol. 9, pp. 115052-115064, 2021.

[6] W. Ali and S. Al-Saadawi, "A deep learning approach for real-time litter detection using convolutional neural networks," *Sustainable Cities and Society*, vol. 81, p. 103754, 2022.

[7] Y. Chen and X. Yang, "A CNN-Based Method for Automatic Litter Detection in Surveillance Videos," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1-8, 2021.

[8] S. Du, B. Wang, and L. Tan, "A deep learning-based approach for litter detection and classification using convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 1, pp. 425-435, 2020.

[9] M. Elhalaby and S. Al-Saadawi, "Real-time litter detection using a novel deep learning-based approach," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 1-16, 2022.

[10] M. Garg and S. Agrawal, "Litter detection and classification using convolutional neural networks," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 10, no. 3, pp. 12-17, 2020.

[11] Y. Zhang, X. Li, and P. Wang, "Deep Learning-Based Suspicious Behavior Detection Using Surveillance Video," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3202-3210, 2018.

[12] J. Wang, Q. Liu, P. Zhang, and W. Li, "Action Recognition Using 3D Convolutional Neural Networks in Surveillance Videos," *IEEE Access*, vol. 8, pp. 2849-2861, 2020.

[13] L. Li, Z. Yu, H. Xu, and Y. Lu, "Environmental Monitoring with Image Processing Techniques: A Comprehensive Survey," *Environmental Science and Technology*, vol. 53, no. 4, pp. 2124-2132, 2019.

[14] H. Ghasemi and Z. Naghshbandi, "Litter detection and classification using deep learning," *Multimedia Tools and Applications*, vol. 80, no. 1-2, pp. 145-160, 2021.

[15] M. Hussain and S. M. Khan, "Litter detection and classification using deep learning," *Sustainable Cities and Society*, vol. 72, p. 103119, 2021.

[16] A. D. Khusno and P. W. Pratama, "Litter Detection and Classification Using Convolutional Neural Network," *Journal of Physics: Conference Series*, vol. 1618, no. 1, p. 012049, 2020.

[17] J. Li, J. Wang, and G. Chen, "Litter detection and classification based on deep learning," *Multimedia Systems*, vol. 27, no. 1, pp. 1-10, 2021.

[18] Y. Liu and X. Yang, "A Lightweight Convolutional Neural Network for Real-time Litter Detection," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 10, pp. 3471-3481, 2022.

[19] Partila, P.; Tovarek, J.; Ilk, G.H.; Rozhon, J.; Voznak, M. Deep learning serves voice cloning: How vulnerable are automatic speaker verification systems to spooting trial. IEEE Commun. Mag. 2020, 58, 100–105.

[20] Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2013, 35, 221–231.

[21] Chengping, R.; Yang, L. Three-dimensional convolutional neural network (3D-CNN) for heterogeneous material homogenization. Comput. Mater. Sci. 2020, 184, 109850.

[22] Y. Zhang, Y. Zhu, and Z. Song, "Detecting suspicious behaviors in video surveillance using deep learning," in *2018 IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 2888-2892.

[23] Y. Wang, X. Xu, and B. Song, "3D CNN based human physical activity recognition in videos," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 4780-4783.