Jurnal Mantik, 7 (3) (2023), ISSN 2685-4236 (Online)

Published by: Institute of Computer Science (IOCS)

Jurnal Mantik

Journal homepage: www.iocscience.org/ejournal/index.php/mantik

Convolutional neural network model for early detection of meatballs containing borax

Aryo Michael[1], Juprianus Rusman[2]

[1,2]Teknik Informatika, Universitas Kristen Indonesia Toraja, Indonesia

ARTICLE INFO

Article history:

Received Nov 17, 2023
Revised Nov 19, 2023
Accepted Nov 24, 2023

Keywords:

Convolution Neural Network;
Deep Learning;
Food Safety;
Image Classification;
Meatball Borax.

ABSTRACT

The use of borax in meatballs to improve the texture and durability of meatballs is still rampant. Borax is very dangerous for consumers. Currently, monitoring of meatballs containing borax is done by experts in the laboratory. The public needs to know this information quickly. Therefore, a system is needed that can detect meatballs containing borax in real time. In this study we built a lightweight Convolutional Neutal Network (CNN) model and searched for optimal hyperparameters for the classification of meatballs containing borax. The results show that the proposed model outperforms other models in classifying meatballs containing borax with an accuracy value of 90%.

Corresponding Author:

Aryo Michael,
Teknik Informatika,
Universitas Kristen Indonesia Toraja,
Jl. Nusantara No 12, Tana Toraja, Sulawesi Selatan, 91811, Indonesia
Email: aryomichael@ukitoraja.ac.id

1. INTRODUCTION

The issue of food safety is very important because it is related to public health (Satcher, 2000), the availability of sufficient and safe food will ensure that everyone consumes food with balanced nutrition and avoids disease (Fung et al., 2018). People's practical lifestyle makes fast food a popular choice today, as seen from the increasing number of fast food restaurants and street vendors (Fatikhani & Setiawan, 2019). In Indonesia, meatballs are one of the most popular street foods. Meatballs are made from ground meat made from fish and meat (usually beef). Meat has a very important role in meeting human nutritional needs due to its high protein content and complete and balanced essential amino acids. However, meat also has the potential to spoil quickly if contaminated by bacteria. This contamination not only causes losses to traders but also producers (Rane, 2011). This condition also results in the practice of using hazardous chemicals such as borax as a preservative in various food products as a chewing agent to improve the texture quality of meatballs to make them more chewy and durable (Nurkhamidah, 2017). However, the use of borax in food is very controversial because it can cause adverse health effects, so many countries have banned the use of borax in food (Pratiwi et al., 2020). Meatballs containing borax are very difficult to distinguish by the eyes of consumers (Putri et al., 2023), so a system is needed that can help consumers as a form of security for their own health through early detection of information related to food products containing harmful substances such as borax in real time.

Artificial intelligence through machine learning (ML) can be a solution to manage and analyze complex data related to food ingredients that have increased in recent years (Ranbir et al., 2022). AI-based advanced computer technology creates algorithms that can operate automatically (Elgendy, 2019). Deep learning is an ML algorithm that is commonly used in computer vision technology in various problem areas such as object detection, image classification, speech recognition, and natural language recognition (Chai et al., 2021; Khan et al., 2020), where the advantages of deep learning are the feature extraction and classification processes performed by the algorithm, this makes deep learning suitable for tasks that require automation (Dara & Tumma, 2018). One deep learning algorithm that is widely used for robust and efficient image classification is convolutional neural networks (CNN) (Alzubaidi et al., 2021). However, CNN models require high-quality datasets in order to be trained well and produce accurate classification results (Khan et al., 2020; Hakim & Fadhil, 2021). In addition, complex CNN architectures also require more training data and computational time to train, are more prone to overfitting, and are more difficult to interpret (Anushka et al., 2021). The number of hyperparameters in CNNs that need to be combined appropriately is also a challenge to improve their performance. For this reason, hyperparameter tuning is needed to optimize the selection of the right parameter combination for the classification model (Minarno et al., 2021; Pardede & Purohita, 2022).

Research on the classification and detection of borax in meatballs has been conducted with texture-based and sensory input approaches and different algorithms. Research by S.D Purwanto et al (Purwanto & Santoso, 2017) developed an android-based application using the naïve bayes algorithm with grayscale image variant features to detect meatballs containing borax where the research results showed a result of 82.77%. Research Saputra, et al (Saputra et al., 2019) applied the K-NN method where the features used were taken from color sensors and pH sensors connected to Arduino. The test results show that the K=5 value gives the highest accuracy of 93.33%. Research by A. Michael et al (Michael et al., 2023) applied a combination of CNN and random forest model architecture to classify meatballs containing borax and non-borax where the results showed the highest model performance in the combination of xception and random forest architecture with an accuracy of 88%. Research Pradana, et al (Pradhana et al., 2023) applied the Articial Neural Network (ANN) method to detect borax at certain levels using odor features detected through the TGS gas sensor sensor. The results showed ANN MLP produced an accuracy value of 95%.

The diversity of meatballs made with various types of meat and ingredients is a challenge to develop a classification model that can be applied generally to all types of meatballs. Each type of meatball has different characteristics, so the classification model developed needs to be able to adapt to the characteristics of each type of meatball. In this research, a lightweight CNN model will be built by using different convolutional layers so that the model can be implemented on devices with limited processing resources, such as smartphones, for early detection of meatballs containing borax. Hyperparameter tuning is applied to determine the right combination of hyperparameters for the CNN model built. We also investigate three convolutional neural network (CNN) architectures designed for resource-constrained devices to evaluate the performance of the proposed model.

2. RESEARCH METHOD

2.1. Data Colection

The initial stage carried out in this research is to collect data in the form of images of meatballs containing borax and not containing borax. The meatball samples used were made independently and also took samples of meatballs sold at several street vendors which were later validated using a borax test kit. Meatball samples made independently were mixed together with borax. The ratio of meat : flour : borax is 1 : 1 :

0%, 1 : 1 : 0.8%, 1 : 1 : 2.4% . After image acquisition, the entire image is clipped manually to take the part of the image that will be classified as shown in Figure 1. All clipped datasets are split based on their class. Table 1 shows the dataset distribution for each image class.

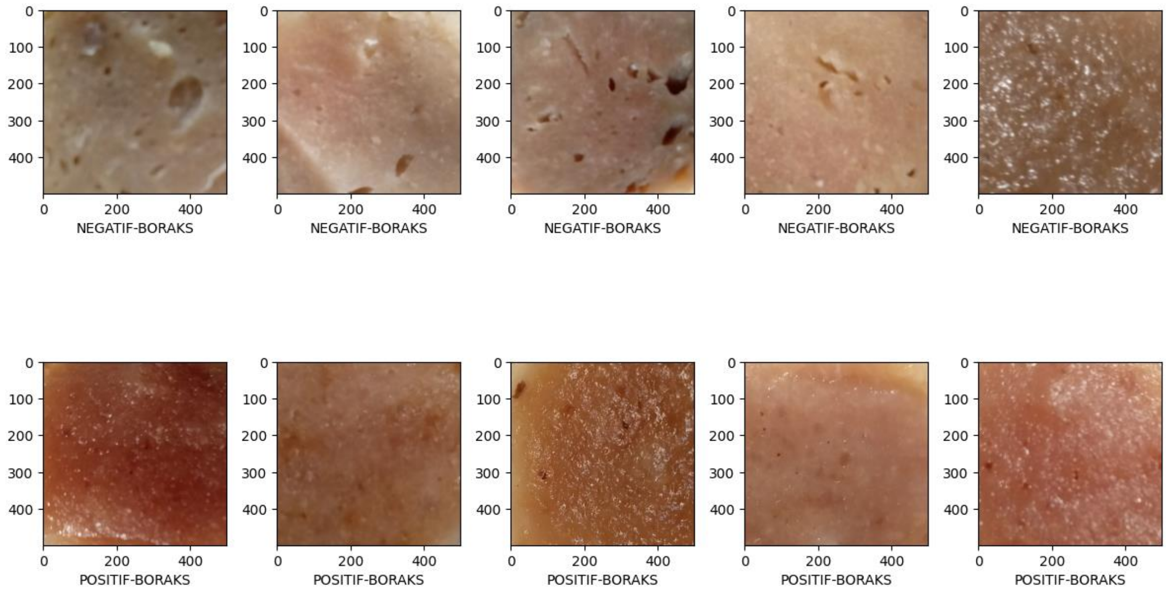


Figure 1. Dataset Image

Table. 1. Distribution of Datasets

Class	Training	Validation	Testing
Negatif-Boraks	600	103	85
Positif-Boraks	618	126	139
Total	1218	229	224

2.2. Preprocessing

Preprocessing is a stage to prepare data so that it can be used as input and processed by the model to be built. Simple image processing techniques include changing the image size to 224x224 pixels, normalization by changing the scale of pixel values to 0–1 intervals, and labeling the image class in numeric form so that it can be processed in the CNN model built.

2.3. Data Augmetation

Data augmentation is a technique to add variety to a dataset without having to add new data. Data augmentation is performed using ImageDataGenerator in the Keras deep learning library. This object is used to perform data augmentation on images during model training. The data augmentation techniques performed are rotating the image (rotation_range) randomly between -30 to 30 degrees, zooming in and out (zoom_range) the image randomly between 0.9 and 1.1 times the original size, shifting the image randomly on the x and y axis by 20% and flipping the image randomly horizontally.

2.4. Proposed CNN Model

This research aims to build a lightweight and high-performance deep learning architecture for the classification of meatballs containing borax. We use Tensorflow to build a CNN architecture as shown in Figure 2. The CNN architecture built consists of five convolution layers arranged sequentially with batch normalization and maxpooling layers. In the fully connected layer, we apply a dense layer by placing a dropout layer.

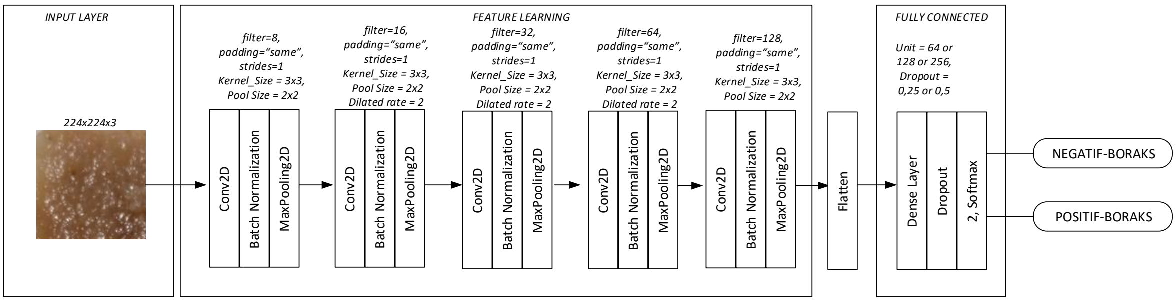


Figure 2. Proposed CNN Architectur

In the convolution layer we apply two types of convolution, standard convolution and dilated convolution with ReLu activation function. In addition to generating feature maps that vary from the input, the use of dilated convolution is to increase the size of the receptive field without increasing the number of parameters (Lin & Wu, 2020) as shown in Figure 3. The activation function ReLu, sets all negative values in the activation map to zero. This effectively removes negative values from the activation map, which makes the network more efficient to train and can improve the performance of the model.

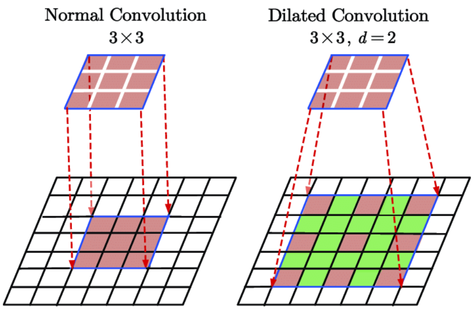


Figure 3. Illustration of Convolution Process with Standard Convolution and Dilated Convolution

We also put a Batch normalization layer which is a proposed technique to improve the performance of CNN. Batch normalization can help stabilize CNN training by homogenizing data distribution and improving training speed(Ioffe & Szegedy, 2015). In addition, batch normalization can help prevent overfitting and gradient dissipation (Long & Zeng, 2019). In the output layer we use 2 neuron units with softmax activation function.

2.5. Hyperparameter Tuning and Training Model

Hyperparameter tuning is a hyperparameter combination search technique that aims to improve model performance. Hyperparameter tuning is performed on "dense layer" and "dropout" as well as "optimizier". The hyperparameter tuning process is performed using the hpparams dashboard. HParams dashboard is a tool that allows you to visualize and manage hyperparameters in machine learning experiments built on TensorFlow.

Table 2. Hyperparameter Configuration for Hyperparameter Tuning

Hyperparameter	Value
Optimizer	Adam, SGD, RMSprop, Nadam
Dropout Fully Connected	0.25, 0.5
Dense layer	128, 256, 512

The next step is to retrain the model using the hyperparameter combination generated in the hyperparameter tuning process. Training is done using a learning rate of 0.001, an epoch of 100, and a batch size of 32. The scheduler function is a callback function used to set the learning rate of the model. This function is called at the beginning of each epoch and returns a new learning rate.

2.6. Model Evaluation

Model performance is based on accuracy score, precision, recall and f1-score (Baştanlar & Ozuysal, 2014). Accuracy score is used to measure whether the model can generally predict correctly. Equation 1 is the formula for calculating the accuracy score. Precision is used to measure whether the model can avoid predicting negative data as positive. Equation 2 is the formula for calculating the precision score. Recall is used to measure whether the model can find all positive data. Equation 3 is the formula for calculating the recall score. f1-score is a combination of accuracy and precision. f1-score can be used to measure overall model performance. Equation 4. is the formula used to calculate the f1-score value.

$$Accurassy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$f1\ score = 2\ x\ \frac{recall\ x\ precision}{recall+precision} \quad (4)$$

3. RESULTS AND DISCUSSIONS

In this study, we train seven models consisting of four proposed models with four hyperparameter combinations that have the best accuracy in the hyperparameter tuning process (SET-1–SET-4), and three CNN architectures that have been trained with the imageNet dataset are also built. We then make adjustments by adding one dense layer, dropout, and model output (SET-5–SET-7) to compare the results of the proposed models and models built for limited resource devices such as MobileNetv2, InceptionV3, and Xception.

Table 3. All Model in Experiment Setting

Set	Model	Dense Layer	Dropout	Optimizier
Set-1	Proposed Model (Best Combination 1)	128	0.25	RMSprop
Set-2	Proposed Model (Best Combination 2)	64	0.5	RMSprop
Set-3	Proposed Model (Best Combination 3)	64	0.5	SGD
Set-4	Proposed Model (Best Combination 4)	128	0.5	Adam
Set-5	MobileNetV2	128	0.25	RMSprop
Set-6	InceptionV3	128	0.25	RMSprop
Set-7	Xception	128	0.25	RMSprop

3.1. Training Model

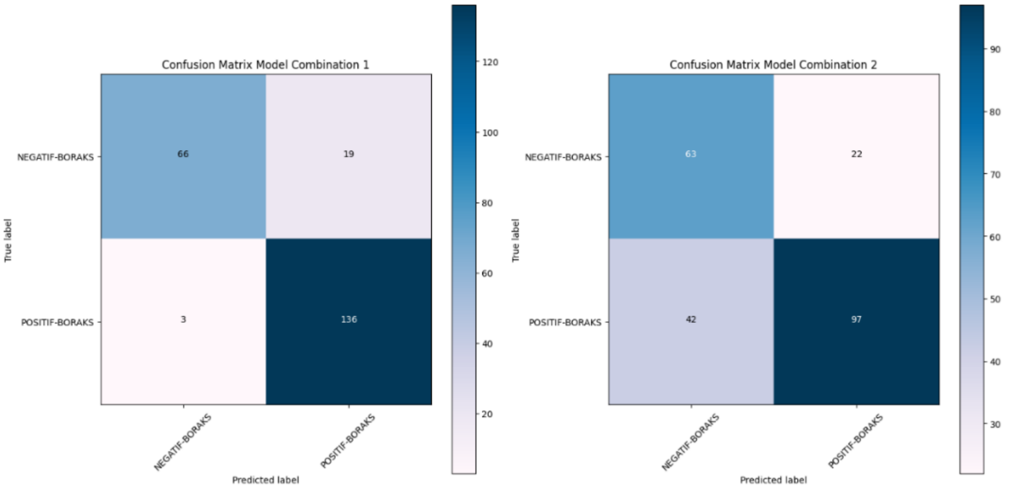
Table 4 shows training result various network structure. The model trained with the combination of hyperparameter tuning results (i.e., SET-1 to SET-4) has relatively reasonable results. The proposed model with combination-1 (Set 1) achieves the highest training and validation accuracy values of 0.95731 and 0.86900, while the models trained using pre-trained models (i.e., SET-5 to SET-7) show less satisfactory results where the validation accuracy are less than 80%.

Table 4. Overall Training Result

Set	Total Params	Trainable Params	Model Size (Mb)	Training		Validation	
				Accurasy	Loss	Accurasy	Loss
Set-1	902578	902082	3.44 MB	0.95731	0.13123	0.86900	0.31698
Set-2	500978	500482	1.91 MB	0.95649	0.12527	0.86026	0.32835
Set-3	500978	500482	1.91 MB	0.92775	0.19794	0.82969	0.38210
Set-4	902578	902082	3.44 MB	0.95567	0.11256	0.89083	0.25599
Set-5	2422210	164226	9.24 MB	0.88834	0.24848	0.77293	0.53527
Set-6	28356770	6553986	108.17 MB	0.87192	0.27265	0.71616	0.57042
Set-7	21124010	262530	80.58 MB	0.88752	0.28394	0.74672	0.48861

3.2. Testing and Evaluation Model

Testing is done on a test dataset to see how well the model recognizes new data. The results of model testing are presented using the confusion matrix in Figure 4.



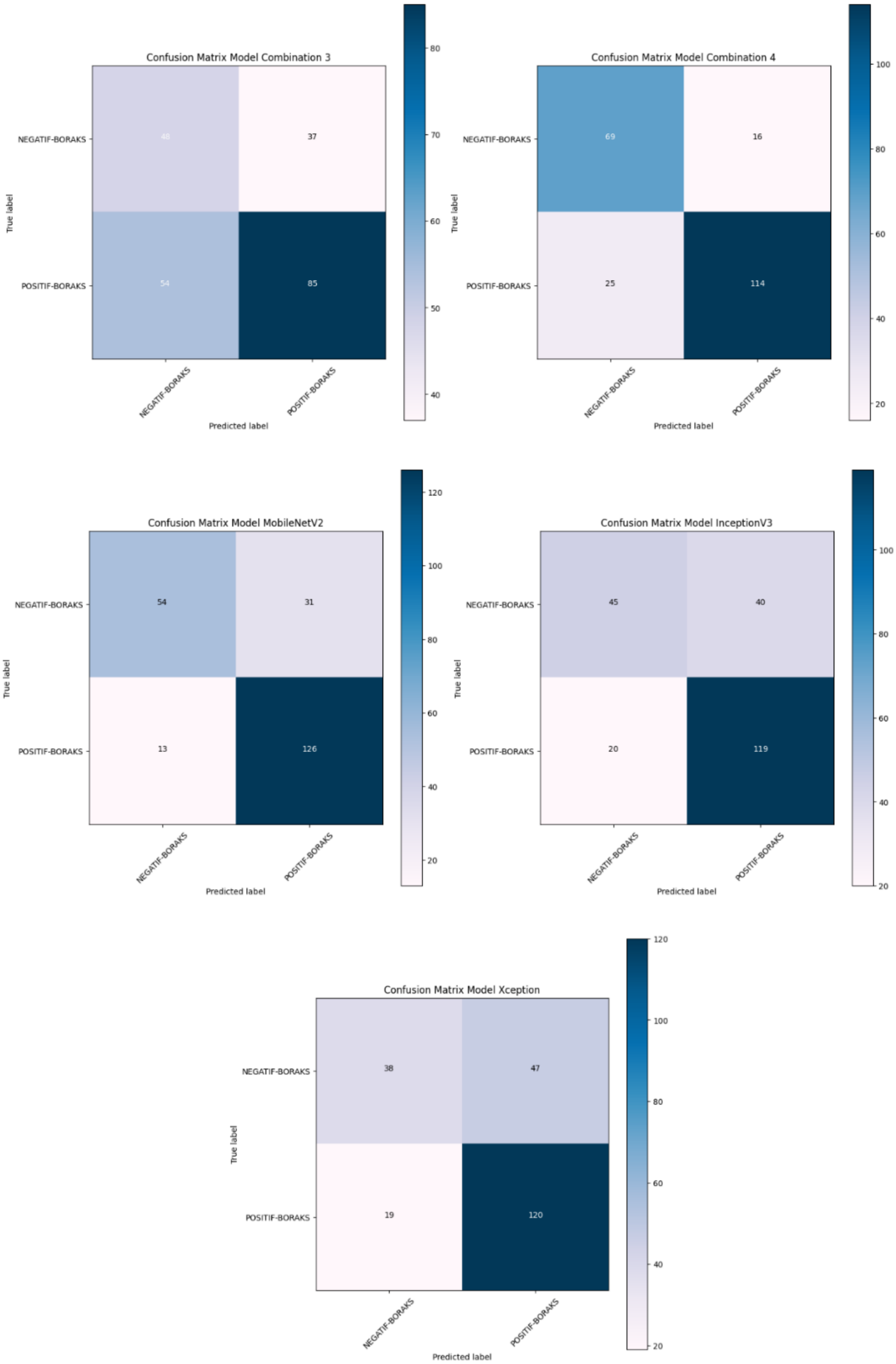


Figure 4. Confution Matrix Overall Model

Testing the proposed models with hyperparameter-1 to hyperparameter-4 combinations (SET-1 - SET-4) and CNN models MobilenetV2 (SET-5), InceptionV3 (SET-6), and Xception (SET-7) shows that all models still have classification errors. The highest misclassification error is found in the proposed model with the combination of hyperparameter-3 (SET-3), which is 91 images out of 224 test datasets, where 37 images are negative-borax and 54 images are positive-borax. The model with the lowest misclassification is the model with the combination of hyperparameter-1 (SET-1), with a misclassification of 22 images out of 224 test datasets, of which 19 images for the negative-borax class and 3 images for the positive-borax class.

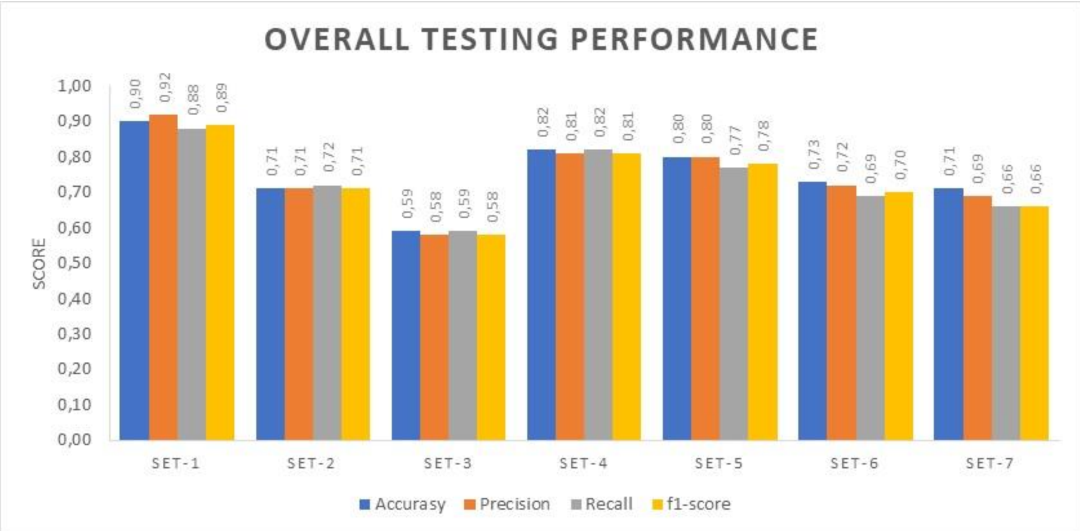


Figure 5. Overall Testing Performance Model

3.3. Discussion

The performance comparison of each model architecture is presented in Figure 10, where the performance of the proposed model with the combination of hyperparameter-1 (SET-1) results in an accuracy score of 0.90, precision of 0.92, recall of 0.88 and f1-score of 0.89. The performance of the model using the hyperparameter-2 (SET-2) combination shows an accuracy score of 0.71, precision of 0.71, recall of 0.72 and f1-score of 0.71. Model performance using a combination of hyperparameter-3 (SET-3) shows an accuracy score of 0.59, precision of 0.58, recall of 0.59 and f1-score of 0.58. Model performance using a combination of hyperparameters-4 (SET-4) shows an accuracy score of 0.82, precision of 0.81, recall of 0.82 and f1-score of 0.81. While the model built using the transfer learning approach, shows the MobileNetV2 (SET-5) model has an accuracy score of 0.80, precision of 0.80, recall of 0.77 and f1-score of 0.78. The InceptionV3 (SET-6) model shows an accuracy score of 0.73, precision of 0.72, recall of 0.649 and f1-score of 0.70. While the Xception model (SET-7) shows performance with an accuracy score of 0.71, precision of 0.69, recall of 0.66 and f1-score of 0.66.

The results of this study indicate that the proposed model with a combination of hyperparameter-1 (SET-1) outperforms the use of other architectures even though it is trained with a smaller number of parameters. We found that using a variety of convolution techniques as well as the right hyperparameter combination can improve the performance of the model. Based on these results, we believe that the proposed model is suitable for the classification of meatballs containing borax. The proposed CNN model is a lightweight CNN model that can be applied to devices that have limited resources such as smartphones so that consumers who will consume meatballs can perform early detection to find out information on borax content in meatballs.

4. CONCLUSION

In this paper, a new technique is developed to classify meatballs containing borax using a lightweight architecture that utilizes a combination of standard convolution and dilated convolution techniques. Experimental results show that the proposed model using a combination of hyperparameters optimized by hyperparameter tuning techniques produces better accuracy compared to models retrained from MobileNetv2, InceptionV3, and Xception. The proposed technique achieves a classification accuracy of 0.90 or 90% with better results when compared to using models that have been trained with tuning on fully connected layers. The lightweight CNN model can be used to develop an early detection application for meatballs containing borax that can be run on devices that have limited resources such as smartphones. Suggestions for future research are that a variety of meatball datasets with different types of ingredients need to be added so that the model built can be used on all types of meatballs. The use of other classification algorithms and other hyperparameter tuning techniques also need to be done.

ACKNOWLEDGEMENTS

This research was funded through the Kemenristekdikti Research Grant Year 2023 based on Decree Number 185/E5/PG.02.00.PL/2023. Our gratitude goes to Kemeristekdikti and all those who have helped so that this research can be completed.

REFERENCES

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Anushka, Arya, C., Tripathi, A., Singh, P., Diwakar, M., Sharma, K., & Pandey, H. (2021). Object Detection using Deep Learning: A Review. *Journal of Physics: Conference Series*, *1854*(1). https://doi.org/10.1088/1742-6596/1854/1/012012

Baştanlar, Y., & Ozuysal, M. (2014). Introduction to Machine Learning Second Edition. In *Methods in molecular biology (Clifton, N.J.)* (Vol. 1107). https://doi.org/10.1007/978-1-62703-748-8_7

Chai, J., Zeng, H., Li, A., & Ngai, E. W. T. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, *6*(March), 100134. https://doi.org/10.1016/j.mlwa.2021.100134

Dara, S., & Tumma, P. (2018). Feature Extraction by Using Deep Learning: A Survey. *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018*, *Iceca*, 1795–1801. https://doi.org/10.1109/ICECA.2018.8474912

Elgendy, M. (2019). *Deep Learning for Vision Systems*. Manning Publications.

Fatikhani, D. A., & Setiawan, A. (2019). The relationship between the level of knowledge regarding fast food and the dietary habits among adolescents in Jakarta, Indonesia. *Enfermeria Clinica*, *29*, 172–175. https://doi.org/10.1016/j.enfcli.2019.04.025

Fung, F., Wang, H. S., & Menon, S. (2018). Food Safety in the 21st Century. *Biomedical Journal*, *41*(2), 88–95. https://doi.org/10.1016/j.bj.2018.03.003

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, 448–456.

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, *53*(8), 5455–5516. https://doi.org/10.1007/s10462-020-09825-6

Lin, Y., & Wu, J. (2020). A Novel Multichannel Dilated Convolution Neural Network for Human Activity Recognition. *Mathematical Problems in Engineering*, *2020*.

https://doi.org/10.1155/2020/5426532
Long, M., & Zeng, Y. (2019). Detecting iris liveness with batch normalized convolutional neural network. *Computers, Materials and Continua*, *58*(2), 493–504. https://doi.org/10.32604/cmc.2019.04378
Michael, A., Palelleng, S., Devi Damayanti, I., & Rusman, J. (2023). Kombinasi Pretrained Model dan Random Forest Pada Klasifikasi Bakso Mengandung Boraks dan Non-Boraks Berbasis Citra. *Teknika*, *12*(1), 27–32. https://doi.org/10.34148/teknika.v12i1.591
Minarno, A. E., Hazmi Cokro Mandiri, M., Munarko, Y., & Hariyady, H. (2021). Convolutional Neural Network with Hyperparameter Tuning for Brain Tumor Classification. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, *6*(2 SE-). https://doi.org/10.22219/kinetik.v6i2.1219
Nurkhamidah, S. (2017). Identifikasi Kandungan Boraks Dan Formalin Pada Makanan Dengan Menggunakan Scientific Vs Simple Methods. *Sewagati*, *1*(1), 26. https://doi.org/10.12962/j26139960.v1i1.2985
Pradhana, A. A. S., Astuti, S. D., Fauziah, Permatasari, P. A. D., Agustina, R., Yaqubi, A. K., Setyawati, H., Winarno, & Putra, C. D. (2023). Sensor Array System Based on Electronic Nose to Detect Borax in Meatballs with Artificial Neural Network. *Journal of Electrical and Computer Engineering*, *2023*, 1–10. https://doi.org/10.1155/2023/8847929
Pratiwi, Y. S., Rini, D. M., Munarko, H., Prayitno, H., & Endah, R. W. (2020). Borax as a Non-Food Grade Addivite in the Perspective of Foof Safety and Human Resources Formation: A Literature Review. *1st International Conference Eco-Innovation in Science, Engineering, and Technology*, *2020*, 119–126. https://doi.org/10.11594/nstp.2020.0518
Purwanto, S. D., & Santoso, I. B. (2017). Sistem Identifikasi Kandungan Boraks pada Bakso Daging Sapi Berbasis Android Menggunakan Algoritma Naive Bayes Classifier. *Matics*, *9*(1), 33. https://doi.org/10.18860/mat.v9i1.3954
Putri, D. P., Yulianti, L. E., Susanti, N. D., & A Putri, S. K. D. F. (2023). Detection of borax on snacks for school children (cilok and siomay) in subang district. *IOP Conference Series: Earth and Environmental Science*, *1168*(1). https://doi.org/10.1088/1755-1315/1168/1/012044
Ranbir, Kumar, M., Singh, G., Singh, J., Kaur, N., & Singh, N. (2022). Machine Learning-Based Analytical Systems: Food Forensics. *ACS Omega*, *7*(51), 47518–47535. https://doi.org/10.1021/acsomega.2c05632
Rane, S. (2011). Street Vended Food in Developing World: Hazard Analyses. *Indian Journal of Microbiology*, *51*(1), 100–106. https://doi.org/10.1007/s12088-011-0154-x
Saputra, D. D., Fitriyah, H., & Setiawan, E. (2019). Sistem Klasifikasi Bakso yang Mengandung Boraks dengan Sensor Warna Menggunakan Metode K-Nearest Neighbor Berbasis Arduino. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(10), 10294–10300. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6663
Satcher, D. (2000). Food safety: A growing global health problem. *Jama*, *283*(14), 1817. https://doi.org/10.1001/jama.283.14.1817