

Kombinasi Pretrained Model dan Random Forest Pada Klasifikasi Bakso Mengandung Boraks dan Non-Boraks Berbasis Citra

Aryo Michael^{1*}, Srivan Palelleng², Irene Devi Damayanti³, Juprianus Rusman⁴

^{1,2,3,4}Program Studi Teknik Informatika, Universitas Kristen Indonesia Toraja, Tana Toraja, Sulawesi Selatan
Email: ^{1*}aryomichael@ukitoraja.ac.id, ²srivanpalelleng@ukitoraja.ac.id,
³irenedevidadamayanti@ukitoraja.ac.id, ⁴jprusman@ukitoraja.ac.id

(Naskah masuk: 9 Jan 2023, direvisi: 14 Feb 2023, diterima: 15 Feb 2023)

Abstrak

Makanan memainkan peran penting dalam kelangsungan hidup manusia. Salah satu makanan tradisional yang populer di Indonesia adalah bakso. Penggunaan Bahan Tambahan Makanan (BTM) dalam makanan, terutama makanan olahan, tidak dapat dihindari. Salah satu BTM yang sering digunakan adalah boraks. Penggunaan boraks dalam pembuatan bakso dapat menyebabkan efek samping kesehatan bagi konsumen. Oleh karena itu, penting untuk menentukan apakah suatu produk bakso mengandung borax atau tidak. Artikel ini bertujuan untuk mengusulkan pendekatan kombinasi model yang terdiri dari *pretrained model* sebagai fitur *extractor* dan *Random Forest* digunakan untuk mengklasifikasikan gambar bakso serta melakukan evaluasi terhadap model yang dibangun. Kombinasi parameter yang digunakan pada *Random Forest* menggunakan parameter terbaik dari *tuning hyperparameter GridsearchCV*. Hasil pengujian menunjukkan kombinasi *pretrained model Xception* dan *Random Forest* menghasilkan kinerja dengan akurasi sebesar 85%, presisi sebesar 87%, *recall* sebesar 83% dan *F1-score* sebesar 85%.

Kata Kunci: *Deep Learning, Klasifikasi Citra, Pretrained Model, Random Forest, Boraks.*

Combination of Pretrained Model and Random Forest in Classification of Borax-Containing and Non-Borax-Containing Meatballs Based on Image

Abstract

Food plays an important role in human survival. One traditional food that is popular in Indonesia is meatballs. The use of food additives in food, especially processed food, is inevitable. One food additives that is often used is borax. The use of borax in the production of meatballs can cause health side effects for consumers. Therefore, it is important to determine whether a meatball product contains borax or not. This article aims to propose a combination model approach consisting of a pretrained model as a feature extractor and a Random Forest to classify meatball images and evaluate the performance of the model. The combination of parameters used in the Random Forest uses the best parameters from the GridsearchCV hyperparameter tuning. The test results show that the combination of the Xception pretrained model and the Random Forest produces a performance with an accuracy of 85%, precision of 87%, recall of 83%, and F1-score of 85%.

Keywords: *Deep Learning, Image Classification, Pretrained Model, Random Forest, Borax.*

I. PENDAHULUAN

Makanan memegang peranan penting dalam kelangsungan hidup manusia. Ada banyak jenis makanan yang dapat dikonsumsi manusia, akan tetapi mengonsumsi makanan tidak hanya sekedar memasukkan makanan ke dalam tubuh, hal terpenting untuk diperhatikan adalah makanan harus seimbang secara gizi dan tentunya aman untuk kesehatan. Salah satu makanan tradisional yang digemari oleh masyarakat Indonesia adalah bakso. Bakso merupakan makanan tradisional berbahan dasar daging, Hal ini dapat dilihat dari banyaknya masyarakat yang menjual olahan bakso.

Penggunaan Bahan Tambahan Makanan (BTM) dalam makanan terutama makanan olahan merupakan hal yang tidak dapat dihindari lagi. Salah satu BTM yang sering digunakan adalah Boraks. Senyawa boraks umumnya digunakan sebagai bahan baku dalam pembuatan deterjen dan *antiseptic*. Boraks adalah senyawa kimia natrium tetraborat yang berbentuk kristal putih tidak berbau. Dalam pembuatan makanan olahan bakso, beberapa produsen bakso seringkali menambahkan senyawa boraks sebagai pengawet dan pengikat daging, agar menghasilkan tekstur bakso yang lebih baik dan tahan lama. Penggunaan senyawa boraks dalam pembuatan bakso dapat menyebabkan efek samping kesehatan bagi konsumen, seperti gangguan sistem pencernaan, kerusakan ginjal, dan gangguan fungsi tiroid [1]. Oleh karena itu, penting untuk menentukan apakah suatu produk bakso mengandung senyawa boraks atau tidak.

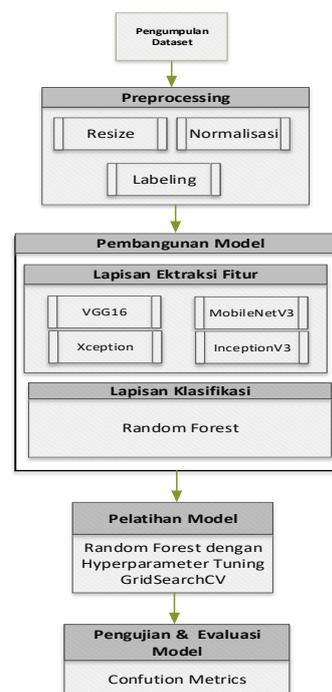
Salah satu cara yang dapat dilakukan adalah dengan menggunakan analisis citra seperti beberapa penelitian yang telah dilakukan terkait topik klasifikasi dan deteksi bakso yang mengandung senyawa boraks [2][3][4][5][6]. Teknologi visi komputer dengan memanfaatkan berbagai algoritma *machine learning* telah banyak digunakan dalam masalah pengelompokan objek melalui fitur tekstur, warna, gerakan dan suara. *Deep learning* (DL) merupakan salah satu cabang dari *machine learning*, yang merupakan bagian dari ilmu komputer yang mempelajari bagaimana cara membuat komputer dapat belajar tanpa diberikan instruksi secara eksplisit. DL memiliki kemampuan untuk memproses data yang sangat besar dan kompleks. Meskipun DL memiliki kemampuan yang luar biasa dalam menangani data yang kompleks dan tidak terstruktur, akan tetapi terdapat beberapa kekurangan pada DL yaitu membutuhkan jumlah data yang sangat besar untuk melakukan pelatihan serta proses pelatihan DL memerlukan waktu yang cukup lama, terutama jika jumlah data yang digunakan sangat besar [7]. Hal ini dapat menyebabkan biaya tinggi dalam pengoperasian sistem DL. Untuk menangani masalah tersebut, metode pembelajaran transfer (*transfer learning*) dapat digunakan. *Transfer learning* adalah metode pembelajaran mesin di mana model yang telah dilatih pada satu tugas dapat digunakan untuk menyelesaikan tugas lain dengan menggunakan sebagian atau seluruh representasi yang diperoleh dari tugas sebelumnya [8], sehingga akan menghemat waktu dan sumber daya yang dibutuhkan untuk melatih model. Salah satu pendekatan dalam *transfer learning* adalah penggunaan *pre-trained model*

sebagai fitur ekstraktor [9]. Proses konvolusi yang ada pada DL akan menghasilkan peta fitur yang akan diproses pada lapisan klasifikasi. Secara umum lapisan klasifikasi *pre-trained model* menggunakan *neural network*. Penyesuaian dilakukan untuk jumlah kelas yang akan diklasifikasi ataupun perubahan algoritma klasifikasi dengan algoritma klasifikasi yang lain pada lapisan ini.

Salah satu algoritma *machine learning* yang banyak digunakan pada berbagai permasalahan klasifikasi pada data tidak terstruktur seperti gambar, audio, dan teks adalah Random Forest [10][11]. Random Forest merupakan salah satu algoritma *ensemble learning* yang bertipe *bootstrap aggregation (bagging)* [12]. Teknik ini bekerja dengan membangun banyak model yang independen dari satu sama lain, kemudian menggabungkan hasil dari masing-masing model tersebut untuk menghasilkan suatu model akhir. Salah satu keuntungan dari *bagging* adalah bahwa ia dapat mengurangi *overfitting* pada model *machine learning*, karena setiap model hanya dipelajari pada sebagian kecil dari data asli, maka tidak akan terlalu tergantung pada pola-pola spesifik dari data tersebut. Dengan demikian, model akhir *bagging* akan lebih *general* dan lebih stabil dibandingkan dengan model pembelajaran mesin yang hanya dipelajari pada data asli.

Berdasarkan hal latar belakang yang telah dikemukakan, maka pada penelitian ini akan mengevaluasi kinerja beberapa *pretrained model* yang akan digunakan sebagai fitur ekstraktor dan mengganti lapisan *fully connected* atau lapisan klasifikasi pada *pre-trained model* menggunakan algoritma Random Forest dalam mengklasifikasi citra bakso mengandung senyawa boraks.

II. METODOLOGI PENELITIAN



Gambar 1. Tahapan Penelitian

Adapun tahapan yang dilakukan dalam penelitian ini disajikan pada Gambar 1.

A. Pengumpulan Data

Citra yang digunakan pada penelitian ini berupa *dataset* citra bakso yang mengandung boraks dan tidak mengandung boraks. Pengambilan citra dilakukan dengan menerapkan berbagai variasi pencahayaan, sudut bentuk dan resolusi.

B. Preprocessing

Preprocessing dilakukan untuk mempersiapkan data agar dapat diolah oleh pada arsitektur yang akan digunakan. Fitur yang dihasilkan untuk klasifikasi sangat dipengaruhi oleh citra *inputan* penyamaan ukuran citra menjadi 224 x 224 piksel, Normalisasi citra dilakukan mengubah nilai piksel pada setiap citra ke interval 0-1, normalisasi dilakukan dengan membagi setiap nilai piksel dengan 255, selanjutnya dilakukan pelabelan pada masing-masing kelas menggunakan teknik *encode*.

C. Pembangunan Model

Model yang dibangun menggunakan metode *transfer learning* dengan pendekatan *pretrained as feature extractor* dimana model yang telah dilatih sebelumnya digunakan sebagai fitur ekstraktor dengan tidak tidak mengikutsertakan lapisan *fully-connected* pada model. Bobot pada setiap *layer* tidak diperbaharui dengan melakukan cara membekukan (*freeze*) pada lapisan fitur *extractor*, selanjutnya dilakukan penyesuaian pada lapisan klasifikasi (*fully connected*) dengan mengganti lapisan klasifikasi yang baru menggunakan algoritma Random Forest. Pada pendekatan ini pelatihan pada model hanya dilakukan pada lapisan klasifikasi untuk melatih algoritma sehingga dapat mengklasifikasi *dataset* yang baru. Pada penelitian ini, menggunakan *pretrained model* yang sudah terkenal yaitu VGG16, MobileNetV3, InceptionV3, Xception.

D. Training Model dengan Hyperparameter Tuning GridSearchCV

Dalam pengembangan model klasifikasi pada *machine learning*, setiap algoritma memiliki seperangkat *hyperparameter*, kombinasi *hyperparameter* yang tepat akan menghasilkan model klasifikasi yang memiliki kinerja yang baik [13]. *Hyperparameter tuning* merupakan proses untuk menemukan kombinasi *hyperparameter* yang paling optimal pada algoritma untuk meningkatkan kinerja model pada *machine learning* [13][14]. Salah satu metode untuk melakukan *tuning hyperparameter* adalah *GridsearchCV* [15]. *GridsearchCV* mengacu pada penggunaan *grid search* dan *cross validation* secara bersamaan untuk mendapatkan indeks evaluasi model klasifikasi. *Grid search* bekerja dengan melakukan pelatihan secara berulang kali sebanyak parameter yang ditentukan sedangkan *cross validation* untuk mengevaluasi kinerja dari kombinasi *hyperparameter* yang telah ditentukan [16]. Nilai dari *hyperparameter* pada setiap algoritma *machine learning* tidak dapat ditentukan dari data, nilai *hyperparameter* harus ditetapkan sebelum sebuah model

menjalani proses pembelajarannya [17]. Beberapa *hyperparameter* yang digunakan untuk mengontrol proses pelatihan pada Random Forest yaitu *n_estimator* merupakan *hyperparameter* yang menentukan jumlah pohon keputusan yang akan dibuat, *max_depth* yang menyatakan seberapa dalam lapisan pada setiap pohon keputusan yang akan dibuat, *min_samples_split* merupakan sampel minimum yang digunakan untuk membuat pemisahan *node* dan *min_samples_leaf* merupakan jumlah sampel minimum untuk setiap *leaf node* [18]. Tabel 1. merupakan *hyperparameter* pada Random Forest yang akan dilakukan *tuning* menggunakan *GridsearchCV*.

Tabel 1. Konfigurasi Parameter Random Forest Pada GridSearchCV

<i>Hyperparameter</i>	<i>Value</i>
<i>bootstrap</i>	<i>True</i>
<i>n_estimator</i>	100, 200, 500, 1000
<i>max_depth</i>	50, 60, 100
<i>min_samples_split</i>	8, 10, 12
<i>min_samples_leaf</i>	3, 4, 5
<i>max_features</i>	2, 3, 10, 100

Dari konfigurasi kombinasi *hyperparameter Gridsearch* pada Tabel 1. akan terbentuk 432 kandidat kombinasi *hyperparameter* pada Random Forest. Pelatihan dilakukan menggunakan *5-fold cross validation* dengan melihat *mean cross validation* sebagai indikator kombinasi *hyperparameter* terbaik.

E. Pengujian dan Evaluasi Model

Pengujian dilakukan terhadap *dataset testing* untuk mengevaluasi kinerja dari model yang dibangun. Salah satu metode yang digunakan untuk pengukuran kinerja dari model klasifikasi adalah *confusion matrix*. *Confusion matrix* dapat digunakan untuk membantu mengevaluasi performa model dengan 4 metrik yaitu akurasi, presisi, *recall* dan *f1-score* [19], [20]. Untuk menghitung akurasi, presisi, *recall* dan *f1-score* menggunakan rumus

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

III. HASIL DAN PEMBAHASAN

Pada penelitian ini, implementasi dilakukan dengan menggunakan bahasa pemrograman *Python* dengan bantuan alat *Jupyter Notebook* dengan memanfaatkan pustaka pengolahan citra dan *machine learning* yang telah tersedia di *Python*.

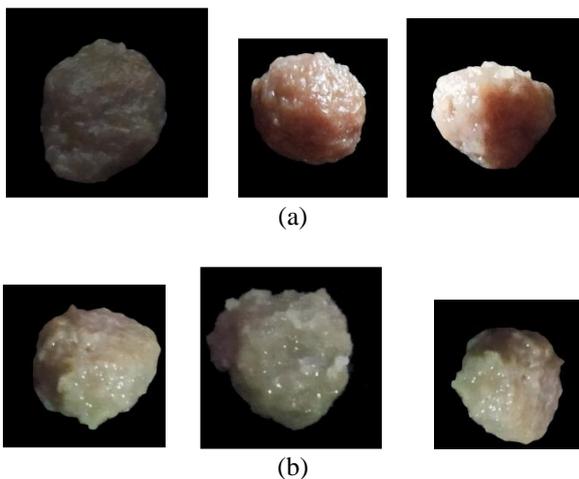
A. Pengumpulan Data

Dataset yang telah dikumpulkan berjumlah 240 citra, dimana masing-masing kelas terdiri atas 120 citra. selanjutnya dibagi ke dalam 2 folder dimana *train* dan *test* dengan persentase 80% sebagai data *training* dan 20% dijadikan sebagai data *testing* seperti pada Tabel 2.

Tabel 2. Pembagian Dataset

Dataset	Kelas	Jumlah
Train	Boraks	96
	Non-Boraks	96
Test	Boraks	24
	Non-Boraks	24

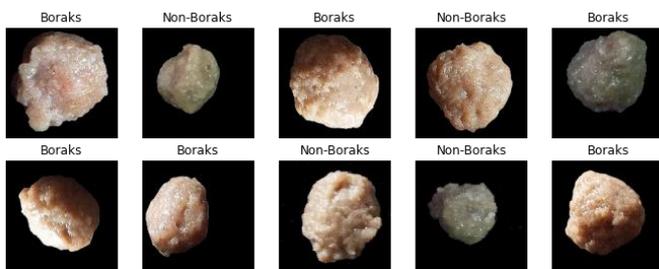
Gambar 2. merupakan sampel dataset citra bakso yang mengandung boraks dan non-boraks



Gambar 2. (a) Citra Bakso Boraks (b) Citra Bakso Non-Boraks

B. Preprocessing

Selanjutnya dilakukan *Preprocessing* seperti yang sudah dijelaskan sebelumnya. Gambar 3. merupakan dataset citra hasil pra proses.



Gambar 3. Dataset Citra Hasil Preprocessing

C. Pembangunan Model

Proses selanjutnya melakukan ekstraksi fitur menggunakan *pretrained model*. *Pretrained model* yang digunakan menggunakan bobot (*weight*) dari dataset

imagenet dengan tidak menyertakan lapisan *fully connected* pada model.

```
from keras.utils import plot_model
Model_MobileNetV3 = MobileNetV3(weights='imagenet', include_top=False, input_shape=(img_size_224p, img_size_224p, 3))
Model_MobileNetV3.trainable = False

##mengambil feature pada data training dan data testing dengan menggunakan pretrained MobileNetV3
feature_extractor=Model_MobileNetV3.predict(X_train_norm)
X_test_feature = Model_MobileNetV3.predict(X_test_norm)

##melihat dimensi fitur yang dihasil oleh pretrained model
feature_extractor.shape
X_test_feature.shape

##melihat fitur pada data training dan data testing yang dihasil oleh pretrained model
print(feature_extractor)
print(X_test_feature)
```



```
[[[ [2.65085697e-01 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
7.45291889e-01 0.00000000e+00
1.62375987e-01 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
5.85655451e-01 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
6.62690103e-01 0.00000000e+00
...
[0.00000000e+00 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
6.61845446e-01 0.00000000e+00
[0.00000000e+00 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
7.26567566e-01 0.00000000e+00
[0.00000000e+00 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
7.32755423e-01 0.00000000e+00]
...
[[ [7.69116044e-01 0.00000000e+00 1.96994364e-01 ... 0.00000000e+00
6.62642062e-01 0.00000000e+00
[5.09858131e-01 0.00000000e+00 1.02025437e+00 ... 0.00000000e+00
2.99177676e-01 0.00000000e+00
[3.68011862e-01 0.00000000e+00 1.25189710e+00 ... 0.00000000e+00
1.93331838e-01 0.00000000e+00]
...
[[ [1.34901118e+00 0.00000000e+00 1.24255431e+00 ... 0.00000000e+00
3.53246152e-01 0.00000000e+00]
```

Gambar 4. Potongan Code dan Hasil Ekstraksi Fitur Dengan Pretrained Model

Setelah proses ekstraksi fitur selesai dilakukan, fitur data *training* yang dihasilkan *reshape* untuk dijadikan *input* pada *Random Forest* yang ditambahkan menggunakan modul *RandomForestClassifier* yang ada pada pustaka *scikit learn*.

D. Training Model Dengan Hyperparameter Tuning GridSearchCV

Sebelum melakukan pelatihan pada model yang dikembangkan, perlu melakukan konfigurasi *hyperparameter* seperti pada Tabel 1. Untuk melakukan pelatihan dengan *GridSearchCV*, digunakan modul *model selection* dari *scikit learn* berbagai macam alat dan metode yang dapat Anda gunakan untuk melakukan pemilihan model dan evaluasi performa model.

Tabel 3. Konfigurasi Parameter Random Forest pada GridSearchCV

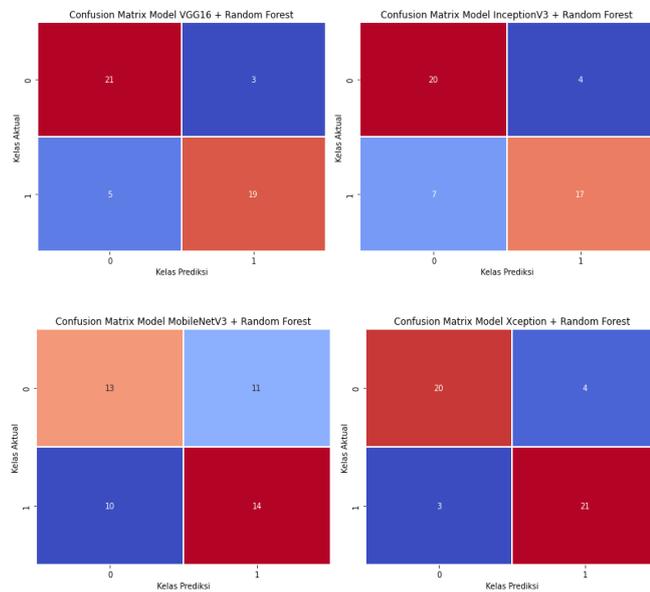
Pretrained model	Best Hyperparameter	Mean Test Score
VGG16	<i>n_estimator</i> : 500	0,771525
	<i>max_depth</i> : 100	
	<i>min_samples_split</i> : 8	
	<i>min_samples_leaf</i> : 4	
MobileNetV3	<i>n_estimator</i> : 200	0,760999
	<i>max_depth</i> : 50	
	<i>min_samples_split</i> : 10	
	<i>min_samples_leaf</i> : 4	
InceptionV3	<i>max_features</i> : 2	0,818084
	<i>n_estimator</i> : 10	

	<u>max_depth: 50</u>	
	<u>min_samples_split: 10</u>	
	<u>min_samples_leaf: 5</u>	
	<u>max_features: 3</u>	
	<u>n_estimator: 500</u>	
	<u>Max_depth: 100</u>	
Xception	<u>min_samples_split: 12</u>	0,802699
	<u>min_samples_leaf: 3</u>	
	<u>max_features: 100</u>	

Hasil pelatihan model dengan metode *GridSearchCV* pada Tabel 3. menunjukkan nilai evaluasi *training* setiap kombinasi *pretrained model* dan *Random Forest* berada pada rentang nilai 0,76 – 0,81.

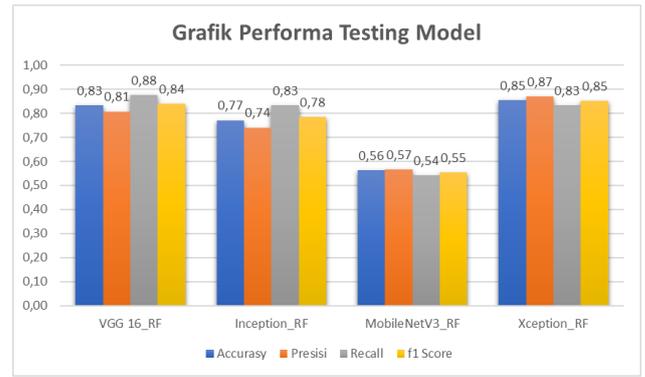
E. Pengujian dan Evaluasi Model

Pada tahap ini, model yang telah dibangun akan diuji menggunakan 48 citra *testing* yang telah disiapkan. Hasil pengujian setiap model disajikan dengan *confusion metrics* pada Gambar 5 menunjukkan bahwa jumlah citra yang diklasifikasikan dengan benar pada model *Xception* dan *Random Forest* paling tinggi dibandingkan dengan 3 model lainnya.



Gambar 5. *Confusion Matrix* Pengujian Model

Setelah pengujian dilakukan pada beberapa model, evaluasi kinerja masing-masing model dilakukan dengan mengukur 4 metrik: akurasi, presisi, *recall* dan *F1-score*.



Gambar 6. Performa Pengujian Model

Gambar 6 menunjukkan model *Xception* dan *Random Forest* memiliki performa yang paling baik untuk metrik akurasi yaitu sebesar 85% sedangkan pada *VGG16* dan *Random Forest*, *InceptionV3* dan *Random Forest*, *MobileNetV3* dan *Random Forest* masing-masing memiliki akurasi sebesar 83%, 77%, dan 56%. Hal yang sama juga ditunjukkan pada nilai presisi dan *F1-score* setiap model dimana model *Xception* dan *Random Forest*. Sedangkan pada nilai *recall*, model *VGG16*, dan *Random Forest* memiliki nilai terbaik yaitu sebesar 88%. Dengan demikian secara keseluruhan berdasarkan pengukuran kinerja setiap model, kombinasi *Xception* dan *Random Forest* memiliki kinerja yang paling baik dalam melakukan klasifikasi bakso yang mengandung boraks

IV. KESIMPULAN

Hasil penelitian menunjukkan dari empat kombinasi *pre-trained model* sebagai ekstraksi fitur dan *Random Forest* sebagai lapisan klasifikasi menunjukkan kombinasi *Xception* dan *Random Forest* memiliki kinerja yang paling baik dalam melakukan klasifikasi citra bakso yang mengandung boraks dan tidak mengandung boraks dari 48 data *testing* yang diuji, dimana kinerja model menghasilkan akurasi sebesar 85%, presisi sebesar 87%, *recall* sebesar 83%, dan *F1-score* sebesar 85%.

REFERENSI

- [1] S. Istiqomah, M. B. Sudarwanto, and E. Sudarnika, "Penambahan Boraks dalam Bakso dan Faktor Pendorong Penggunaannya Bagi Pedagang Bakso di Kota Bengkulu," *J. Sain Vet.*, vol. 34, no. 1, p. 1, 2017, doi: 10.22146/jsv.22806.
- [2] D. D. Saputra, H. Fitriyah, and E. Setiawan, "Sistem Klasifikasi Bakso yang Mengandung Boraks dengan Sensor Warna Menggunakan Metode K-Nearest Neighbor Berbasis Arduino," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 10, pp. 10294–10300, 2019, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6663>
- [3] I. B. Santoso, "Deteksi Boraks Pada Bakso Berbasis Image Dengan Menggunakan Gaussian Classifier,"

- Matics*, vol. 7, no. 1, p. 27, 2015, doi: 10.18860/mat.v7i1.2873.
- [4] N. Fitriya, Shabri Putra Wirman, and Riri Gusfita, "Rancang Bangun Alat Pendeteksi Kontaminan Borak Pada Makanan Berbasis Android," in *Prosiding Saintek Semnas MIPAKes Umri*, 2021, vol. 2, pp. 232–241. [Online]. Available: <https://ejurnal.umri.ac.id/index.php/Semnasmipakes/article/view/2881/1604>
- [5] D. G. Kusumafikri, A. Muid, and I. Sanubary, "Rancang Bangun Alat Pendeteksi Bakso Mengandung Boraks Menggunakan Sensor Resistansi," *Prism. Fis.*, vol. 7, no. 2, pp. 114–119, 2019, doi: 10.26418/pf.v7i2.34139.
- [6] A. Wahyudi, "Sistem Pendeteksi Boraks (Na2b4o7 10 H2o) Pada Bakso Daging Sapi Berdasarkan Image Dengan Menggunakan Metode Jaringan Saraf Tiruan LVQ (Learning Vector Quantization)," *Universitas Islam Negeri Maulana Malik Ibrahim*, 2016. [Online]. Available: <http://etheses.uin-malang.ac.id/3820/1/12650114.pdf>
- [7] M. Saputra, Kusrini, and M. P. Kurniawan, "Identifikasi Mutu Biji Kopi Arabika Berdasarkan Cacat Dengan Teknik Convolutional Neural Network," *Inspir. J. Teknol. Inf. dan Komun.*, vol. 10, no. 1, pp. 27–35, 2020.
- [8] A. Michael, "Komparasi Kombinasi Pre-trained Model dengan SVM pada Klasifikasi Kematangan Kopi Berbasis Citra," *J. Dyn. Saint*, vol. 7, no. 1, pp. 42–48, 2022, doi: <https://doi.org/10.47178/dynamicsaint.v7i1>.
- [9] M. Elgendy, *Deep Learning for Vision Systems*. Manning Publications, 2019.
- [10] N. Wuryani and S. Agustiani, "Random Forest Classifier untuk Deteksi Penderita COVID-19 berbasis Citra CT Scan," *J. Tek. Komput. AMIK BSI*, vol. 7, no. 2, pp. 187–193, 2021, doi: 10.31294/jtk.v4i2.
- [11] J. Media and I. Budidarma, "Klasifikasi Dialek Pengujar Bahasa Inggris Menggunakan Random," *J. Media Inform. Budidarma*, vol. 5, no. April, pp. 439–446, 2021, doi: 10.30865/mib.v5i2.2754.
- [12] U. Khultsum and A. Subekti, "Penerapan Algoritma Random Forest dengan Kombinasi Ekstraksi Fitur Untuk Klasifikasi Penyakit Daun Tomat," *J. Media Inform. Budidarma*, vol. 5, no. 1, p. 186, 2021, doi: 10.30865/mib.v5i1.2624.
- [13] W. Nugraha and A. Sasongko, "Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search," *Sist. J. Sist. Inf.*, vol. 11, no. 2, pp. 391–401, 2022.
- [14] H. Sastypratiwi, H. Muhardi, and M. Noveanto, "Klasifikasi Emosi Pada Lirik Lagu Menggunakan Algoritma Multiclass SVM dengan Tuning Hyperparameter PSO," *J. Media Inform. Budidarma*, vol. 6, no. 4, pp. 2279–2286, 2022, doi: 10.30865/mib.v6i4.4609.
- [15] R. Ghawi and O. Access, "Efficient Hyperparameter Tuning with Grid Search for Text Categorization using kNN Approach with BM25 Similarity," *Open Comput. Sci.*, vol. 9, no. 1, pp. 160–180, 2019, doi: <https://doi.org/10.1515/comp-2019-0011>.
- [16] T. Yan, S.-L. Shen, A. Zhou, and X.-S. Chen, "Prediction of geological characteristics from shield operational parameters using integrating grid search and K-fold cross validation into stacking classification algorithm," *J. Rock Mech. Geotech. Eng.*, vol. 14, no. 3, 2022, doi: 10.1016/j.jrmge.2022.03.002.
- [17] T. Yu and H. Zhu, "Hyper-Parameter Optimization : A Review of Algorithms," 2020. doi: <https://doi.org/10.48550/arXiv.2003.05689>.
- [18] R. Darmawan, D. Purnomo, S. Nilawati, and Fery Perdiansyah, "Implementasi Algoritma Machine Learning untuk Pengembangan Model Prediktif atas Tingkat Non-Performing Loan dalam rangka Penjaminan Kredit UMKM Pemulihan Ekonomi Nasional," in *SENAKOTA - Seminar Nasional Ekonomi dan Akuntansi*, 2021, vol. 1, no. 1, pp. 97–110. [Online]. Available: <https://prosiding.senakota.nusaputra.ac.id/article/view/15>
- [19] Y. Baştanlar and M. Ozuysal, *Introduction to Machine Learning Second Edition*, vol. 1107. 2014. doi: 10.1007/978-1-62703-748-8_7.
- [20] A. Tharwat, "Classification Assessment Methods," *Appl. Comput. Informatics*, vol. 17, no. 1, pp. 168–192, 2018, doi: 10.1016/j.aci.2018.08.003.