



Penerapan Algoritma YOLOv4-Tiny Dan Efficientnetv2-S Untuk Deteksi Kesegaran Ikan Gurami

Hans Richard Alim Natadja^{1*}, Daniel Martomanggolo Wonohadidjojo²

^{1,2}Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Ciputra, Indonesia

Email: ¹hansrichard2000@gmail.com, ²daniel.m.w@ciputra.ac.id

Abstract

Fish production is one of the largest in Indonesia. Among many fishery products, Gurami fish is one of the most widely processed by society. However, the decrease in the freshness of fishery products is susceptible to occur when they reach the hands of consumers. Several methods for detecting fish freshness have been applied to assist in obtaining fresh fish products. However, some of these methods have a lack of accuracy rate, not all of them were developed to detect gurami freshness. Therefore, a solution is needed to help people detect the freshness of Gurami fish and make it accessible on mobile devices. This solution can be realized by using Deep Learning methods. The methods proposed in this study use the Convolutional Neural Network (CNN) by utilizing the YOLOv4-Tiny algorithm to detect the Region of Interest (ROI) and the EfficientNetV2-S architecture freshness classification on Gurami fish images. The training process using both methods can produce an ROI detection model with a mean average precision of 93,58% and a classification model with an accuracy rate of 93%.

Keywords: Gurami, Deep Learning, CNN, YOLOv4-Tiny, EfficientNetV2-S

Abstrak

Produksi ikan merupakan salah satu yang terbesar di Indonesia. Dari banyak produk perikanan, ikan Gurami menjadi salah satu yang banyak diolah oleh masyarakat. Namun, penurunan kesegaran produk perikanan rentan terjadi ketika sampai ke tangan konsumen. Beberapa metode deteksi kesegaran ikan telah diterapkan untuk membantu mendapatkan produk perikanan yang masih segar. Namun dari beberapa metode tersebut terdapat kekurangan pada tingkat akurasi dan tidak semuanya dikembangkan untuk deteksi kesegaran ikan Gurami. Dari hal tersebut dibutuhkan sebuah solusi yang mampu membantu masyarakat untuk bisa melakukan deteksi kesegaran ikan Gurami dan dapat digunakan pada perangkat seluler. Solusi tersebut dapat diwujudkan dengan menggunakan metode Deep Learning. Metode yang diusulkan dalam penelitian ini menggunakan Convolutional Neural Network (CNN) dengan memanfaatkan kombinasi algoritma YOLOv4-Tiny untuk melakukan deteksi Region of Interest (ROI) dan arsitektur EfficientNetV2-S untuk klasifikasi kesegaran pada citra ikan Gurami. Proses training dengan menggunakan kedua metode tersebut mampu menghasilkan model deteksi ROI dengan mean average precision sebesar 93,58% dan model klasifikasi dengan tingkat akurasi sebesar 93%.

Kata kunci: Gurami, Deep Learning, CNN, YOLOv4-Tiny, EfficientNetV2-S

1. PENDAHULUAN

Produk perikanan merupakan salah satu produk bahan makanan dengan konsumsi terbesar di Indonesia. Ikan Gurami turut menjadi salah satu varietas produk perikanan dengan total produksi yang besar. Data dari Kementerian Kelautan dan Perikanan Indonesia menunjukkan bahwa produksi ikan Gurami mencapai 149.170 ton pada tahun 2021 [1]. Namun adanya perlakuan yang tidak tepat ketika melakukan distribusi dari produk perikanan tersebut menyebabkan penurunan kesegaran pada ikan ketika mencapai tangan konsumen [2]. Di lain sisi kesegaran ikan menjadi salah satu penentu yang membuat seorang konsumen

berminat untuk membeli sebuah produk perikanan [3]. Hal tersebut membuat metode deteksi kesegaran ikan dikembangkan agar konsumen tetap dapat memperoleh manfaat dari produk perikanan.

Salah satu cara untuk melakukan deteksi pada ikan adalah dengan menggunakan metode organoleptik yang mengikuti standar SNI 2729 dengan melakukan pengamatan pada kondisi bagian tubuh ikan menggunakan pancaindra [4]. Walaupun begitu tidak semua masyarakat mampu melakukan penentuan kesegaran ikan secara akurat dengan cara tersebut. Penggunaan teknologi *Computer Vision* menjadi solusi yang dapat digunakan untuk bisa mempermudah dalam melakukan deteksi tingkat kesegaran secara organoleptik pada bahan makanan berupa ikan tanpa perlu melakukan sentuhan fisik yang merusak sebuah ikan. Dari beberapa metode yang diterapkan untuk *Computer Vision*, metode *Deep Learning* dengan menggunakan *Convolutional Neural Network* (CNN) menjadi salah satu metode dengan hasil yang lebih akurat [5].

Metode CNN telah digunakan dalam studi terdahulu untuk melakukan deteksi kesegaran ikan menggunakan parameter mata dan warna insang. Dalam studi terdahulu tersebut digunakan kombinasi model Detectron2 untuk deteksi objek mata dan insang, serta arsitektur CNN GoogLeNet untuk klasifikasi kesegaran ikan dari kondisi dua objek tersebut. Dalam menguji tingkat kesegaran menggunakan mata sebagai *Region of Interest*, sistem telah berhasil mencapai tingkat presisi sebesar 83%, sementara untuk mendeteksi tingkat kesegaran dengan menggunakan insang sistem hanya bisa mencapai pada tingkat presisi sebesar 40% [6]. Penelitian ini menunjukkan bahwa kombinasi algoritma untuk *object detection* dan *image classification* dapat digunakan untuk membantu deteksi kesegaran ikan. Namun penelitian tersebut belum menerapkan algoritma YOLO dan arsitektur CNN terbaru. Salah satu algoritma deteksi objek yang populer saat ini adalah algoritma *You Only Look Once* (YOLO) [7]. Penelitian yang berkaitan dengan implementasi algoritma YOLO di bidang perikanan telah dilaksanakan dengan melakukan kombinasi algoritma YOLOv4-Tiny dengan *Wing Convolution Layer* untuk melakukan deteksi bagian tubuh ikan. Hasil penelitian tersebut mampu menghasilkan varian algoritma YOLOv4 yang mampu bekerja lebih akurat dengan hasil *mAP* sebesar 92.38% [8]. Namun algoritma tersebut tidak dirancang untuk melakukan penentuan tingkat kesegaran ikan Gurami.

Sementara untuk melakukan klasifikasi citra dalam menentukan kesegaran ikan dapat digunakan arsitektur CNN terbaru yang berisi kumpulan lapisan konvolusi yang dapat diatur untuk bisa melakukan klasifikasi sesuai dengan *dataset* yang ada. Penelitian yang berkaitan dengan klasifikasi kesegaran menggunakan CNN juga telah dilakukan dengan memanfaatkan kombinasi arsitektur MobileNetV1 dengan metode *Depthwise Separable convolution bottleneck with Expansion* diimplementasi pada perangkat seluler untuk melakukan klasifikasi kesegaran dengan parameter mata ikan. Namun akurasi yang berhasil dicapai oleh metode tersebut hanya sebesar 63,21% [9]. Salah satu arsitektur CNN terbaru lain yang dapat digunakan untuk klasifikasi citra adalah arsitektur EfficientNetV2 [10].

Namun model Deep Learning yang dihasilkan oleh algoritma YOLO dan EfficientNetV2 membutuhkan perangkat komputer dengan spesifikasi tinggi agar bisa dimanfaatkan untuk melakukan proses deteksi dan klasifikasi citra [5]. Sementara masyarakat lebih banyak menggunakan perangkat seluler dalam aktivitas sehari-hari [11]. Perangkat seluler secara umum tidak memiliki kekuatan komputasi yang besar seperti komputer *desktop*. Di lain sisi proses *training* untuk menghasilkan model Deep Learning menggunakan kedua algoritma tersebut memerlukan jumlah dataset yang banyak dan waktu yang relatif lama apabila dilakukan proses *training* dari awal dengan *resource* komputasi yang seadanya. Sehingga perlu digunakan arsitektur *pre-trained Convolutional Neural Network* (CNN) yang diperkecil untuk mengurangi beban komputasi pada perangkat seluler dengan tetap memberikan hasil deteksi yang akurat dan menghemat waktu dalam melakukan proses *training* model. Karena itu dalam penelitian ini digunakan varian *pre-trained* CNN dari algoritma YOLOv4 yaitu YOLOv4-Tiny dan varian *pre-trained* CNN dari arsitektur EfficientNetV2 yaitu EfficientNetV2-S. Agar kedua *pre-trained* CNN tersebut dapat digunakan untuk melakukan deteksi dan klasifikasi kesegaran pada ROI berupa mata dan insang ikan Gurami digunakan metode *transfer learning*. Metode tersebut digunakan untuk bisa melakukan proses *training* model Deep Learning dalam waktu singkat dengan memanfaatkan pengetahuan dari model yang sudah ada sebelumnya [12]. Sehingga dengan memanfaatkan seluruh metode tersebut dapat dihasilkan sebuah model Deep Learning yang mampu melakukan deteksi ROI mata dan insang ikan Gurami dan model klasifikasi tingkat kesegaran ikan Gurami berdasarkan ROI yang terdeteksi.

2. METODOLOGI PENELITIAN

Dalam penelitian ini digunakan metode eksperimen dengan melakukan pengujian pada berbagai skenario proses *training* model deteksi menggunakan algoritma YOLOv4-Tiny dalam mendeteksi ROI berupa mata dan insang dari ikan Gurami, serta pengujian pada proses *training* model klasifikasi kesegaran ikan Gurami menggunakan arsitektur EfficientNetV2-S. Alur dari tahapan eksperimen untuk menguji kedua algoritma tersebut dapat dilihat pada Gambar 1.



Gambar 1. Alur Penelitian

Pada eksperimen dalam penelitian ini digunakan 4 jenis *dataset* yaitu *dataset* ikan A yang berisi gabungan *dataset* primer dan *dataset* sekunder kumpulan citra ikan yang telah dilakukan anotasi pada ROI berupa mata ikan Gurami, *dataset* ikan B yang berisi kumpulan data primer citra insang ikan Gurami yang telah dianotasi, *dataset* ikan C yang berupa gabungan *dataset* primer dan sekunder dari citra mata

ikan Gurami yang sudah dilakukan *labeling* untuk klasifikasi kesegaran, serta *dataset* ikan D primer yang berisi kumpulan data primer citra insang ikan Gurami yang sudah dilakukan *labeling* untuk klasifikasi kesegaran.

Dataset ikan A dan B digunakan untuk membantu proses *training* pada algoritma YOLOv4-Tiny. Sementara *dataset* ikan C dan D digunakan untuk proses *training* pada arsitektur EfficientNetV2-S. *Dataset* primer yang terkumpul untuk *dataset* ikan A dan B dilakukan proses *image annotation* dan *labeling*, sementara *dataset* primer untuk *dataset* ikan C dan D hanya dilakukan proses *labeling* sebelum digunakan untuk proses *training*. Proses penentuan label pada *dataset* ikan C dan D mengacu pada standar kesegaran ikan yang ditetapkan berdasarkan SNI 2729 tentang Ikan Segar [4].

Kemudian *Dataset* sekunder yang digunakan dalam penelitian ini menggunakan sebagian citra mata ikan dari penelitian yang dilakukan oleh [8] dan sebagian citra mata ikan Gurami dalam penelitian oleh [6]. *Dataset* dari kedua sumber tersebut sudah dilakukan *image annotation* dan *labeling* dalam penelitian sebelumnya.

Setelah seluruh *dataset* yang diperlukan terkumpul dan selesai diolah, tahap berikutnya dilakukan *pre-processing* pada seluruh *dataset* dengan mengubah ukuran seluruh citra pada *dataset* menjadi 640x640 kemudian membagi *dataset* menjadi *train validation test set* dengan rasio pembagian 70:20:10. Setelah itu dilakukan *image augmentation* dengan menggunakan metode *rotation*, *flipping*, *brightness*, *contrast*, dan *exposure* pada seluruh *dataset* dengan tujuan untuk meningkatkan akurasi dari model *Deep Learning*. Khusus untuk *dataset* yang digunakan pada proses *training* EfficientNetV2-S diberi penambahan metode *augmentasi blur*. Proses *image augmentation* hanya diberlakukan pada *train set*.

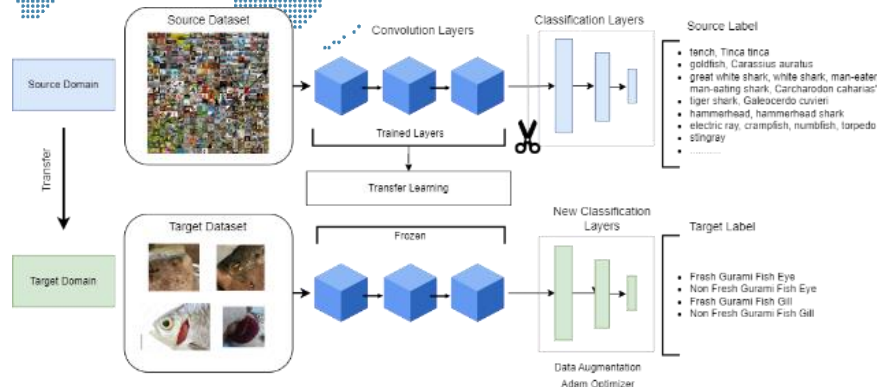
Tahap berikutnya dilakukan proses *training* pada *dataset* dengan memanfaatkan algoritma YOLOv4-Tiny dan EfficientNetV2-S. Proses *training* model deteksi menggunakan algoritma YOLOv4-Tiny memanfaatkan *pre-trained CNN* yang dikembangkan oleh [13]. Proses *training* pada algoritma YOLOv4-Tiny dilaksanakan dengan melakukan eksperimen sebanyak 4 skenario pada parameter *training* yang dapat dilihat pada Tabel 1.

Tabel 1. Skenario *Training* YOLOv4-Tiny

Parameter	Skenario			
	1	2	3	4
<i>Batch</i>	64	64	64	64
<i>Sub divisions</i>	64	64	64	64
Ukuran	416 x 416	416 x 416	608 x 608	608 x 608
<i>Momentum</i>	0.9	0.9	0.9	0,9
<i>Learning Rate</i>	0.00261	0.00261	0.00261	0.00261
<i>Max Batches</i>	6000	10000	6000	10000
<i>Steps</i>	4800, 5400	8000, 9000	4800, 5400	8000, 9000

Sementara proses *training* EfficientNetV2-S memanfaatkan *pre-trained CNN* yang diakses melalui *Keras Library* [14] dan dilaksanakan menggunakan metode

transfer learning, serta *optimizer* Adam dengan rancangan arsitektur yang dapat dilihat pada Gambar 2.



Gambar 2. Metode *Transfer Learning* pada EfficientNetV2-S

Dalam melakukan proses *training* menggunakan arsitektur EfficientNetV2-S juga diterapkan beberapa skenario pengaturan parameter *training* untuk kebutuhan eksperimen. Detail penggunaan skenario *training* pada EfficientNetV2-S dapat dilihat pada Tabel 2 dan Tabel 3.

Tabel 2. Skenario *Training* EfficientNetV2-S dengan *Pooling Layer*

Parameter	Skenario			
	1	2	3	4
Ukuran	416 x 416	416 x 416	416 x 416	416 x 416
<i>Pooling layer</i>	Global Average Pooling2D()	Global Average Pooling2D()	Global Average Pooling2D()	Global Average Pooling2D()
<i>Dense</i>	200	200	256	256
	100	100	128	128
	50	50	64	64
	4	4	4	4
<i>Regularization</i>	Batch Normali-zation Dropout(0.5)	Batch Normali-zation Dropout(0.5)	Batch Normali-zation Dropout(0.5)	Batch Normali-zation Dropout(0.5)
<i>Optimizer</i>	Adam	Adam	Adam	Adam
<i>Learning Rate</i>	0.001	0.00001	0.00001	0.00002

Tabel 3. Skenario *Training* EfficientNetV2-S dengan *Pooling Layer* dan *Reshaping Layer*

Parameter	Skenario			
	5	6	7	8
Ukuran	416 x 416	416 x 416	416 x 416	416 x 416
<i>Pooling layer</i>	-	-	-	Global Average Pooling2D()
<i>Reshaping layer</i>	Flatten()	Flatten()	Flatten()	Flatten()
<i>Dense</i>	200	200	256	256



Parameter	Skenario			
	5	6	7	8
	100 50 4	100 50 4	128 64 4	128 64 4
Regularization	Batch Normali-zation Dropout(0.5)	Batch Normali-zation Dropout(0.5)	Batch Normali-zation Dropout(0.5)	Batch Normali-zation Dropout(0.5)
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.00001	0.00001	0.00002

Berdasarkan Tabel 2 dan Tabel 3, langkah pertama yang dilakukan untuk proses *training* menggunakan EfficientNetV2-S adalah dengan menambahkan lapisan GlobalAveragePooling2D. Langkah kedua dilanjutkan dengan eksperimen untuk penambahan lapisan Dense dengan menggunakan jumlah lapisan dan jumlah neuron sesuai eksperimen skenario 1 hingga 4 pada Tabel 2 dimana seluruh lapisan Dense yang ditambahkan memanfaatkan metode aktivasi ReLU. Lalu diberikan lapisan BatchNormalization dan Dropout sebesar 0.5. Langkah terakhir adalah menambahkan lapisan Dense dengan jumlah neuron 4 dan menggunakan metode aktivasi Softmax untuk melakukan klasifikasi. Jumlah neuron pada lapisan Dense terakhir tersebut ditentukan berdasarkan jumlah label klasifikasi yang ingin diprediksi. Sementara eksperimen dengan skenario 5 hingga 8 dilaksanakan dengan memanfaatkan lapisan Flatten. Kemudian ditambahkan lapisan Dense dengan metode aktivasi ReLU, BatchNormalization dan Dropout sesuai pengaturan pada Tabel 3.

3. HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan hasil proses *training* dan evaluasi dari model yang dihasilkan algoritma YOLOv4-Tiny dan EfficientNetV2-S. Evaluasi pada algoritma YOLOv4-Tiny memanfaatkan grafik *mAP*, *loss*, dan *performance metrics* yang dihitung berdasarkan aturan PascalVOC [15], [16] dari hasil proses *training* terhadap empat skenario yang telah diimplementasikan berdasarkan parameter dalam Tabel 1. Selain itu pada *framework* Darknet, hasil *mAP* yang disajikan berdasarkan *Intersection over Union (IoU) threshold* sebesar 0,5 atau juga disebut dengan *mAP@0.5*. Tabel 4 menunjukkan hasil evaluasi model deteksi.

Tabel 4. Performa Model Deteksi Hasil *Training* YOLOV4-Tiny

Skenario	AP _{mata}	AP _{insang}	mAP@0.5
1	81,74%	96,21%	88.97%
2	87,29%	99,52%	93,41%
3	89,58%	97,52%	93,55%
4	91,73%	95,43%	93,58%

Dari keempat skenario yang diberlakukan dalam proses *training* model deteksi menggunakan algoritma YOLOv4-Tiny dapat disimpulkan bahwa hasil *training* dari skenario keempat menghasilkan model deteksi dengan *mAP@0.5*

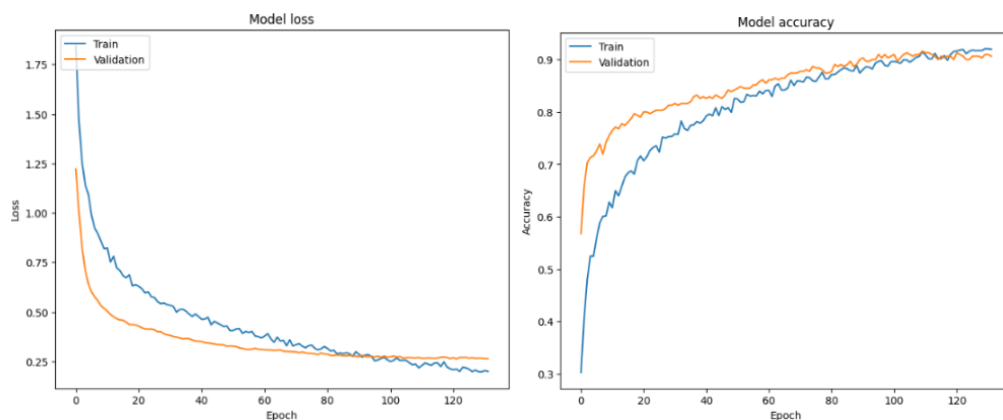
yang paling tinggi yaitu sebesar 93.58%, serta memiliki nilai AP yang lebih seimbang untuk kedua ROI yang ingin dideteksi. Selain itu penambahan total BFLOPS yang ada pada hasil *training* model deteksi menggunakan algoritma YOLOv4-Tiny di skenario keempat masih tergolong rendah. Sehingga model dari hasil *training* pada skenario keempat menjadi yang paling baik untuk digunakan pada aplikasi perangkat seluler.

Evaluasi hasil dari proses *training* model klasifikasi dari arsitektur EfficientNetV2-S, digunakan grafik *loss*, *accuracy*, serta *performance metrics*. Hasil dari proses *training* menggunakan 8 skenario dalam Tabel 2 dan Tabel 3 dapat dilihat pada Tabel 5.

Tabel 5. Komparasi Accuracy dan Loss Hasil Training EfficientNetV2-S

Skenario	Epochs	Final epoch accuracy	Final epoch val_accuracy	Final epoch loss	Final epoch val_loss
1	31	0,9544	0,8903	0,1161	0,3138
2	200	0,8649	0,8484	0,3306	0,3234
3	200	0,8770	0,8742	0,2836	0,2905
4	132	0,9198	0,9065	0,1995	0,2643
5	17	0,9825	0,8581	0,0513	0,5640
6	39	0,9878	0,8806	0,0598	0,2818
7	35	0,9916	0,8903	0,0520	0,2625
8	123	0,9082	0,9032	0,2324	0,2551

Dari Tabel 5 dapat disimpulkan bahwa proses *training* model klasifikasi menggunakan EfficientNetV2-S memberikan hasil terbaik apabila digunakan parameter pada skenario 4. Dari hasil grafik skenario tersebut pada Gambar 3 dapat dilihat bahwa kurva *train* dan kurva *validation* mampu untuk saling bertemu dan berakhir dengan gap yang tidak besar serta memiliki akurasi akhir sebesar 0,9198 dan *loss* yang rendah sebesar 0,1995. Hasil dari kedua kurva dalam grafik hasil *training* tersebut tidak menunjukkan adanya konfisi *underfit* maupun *overfit* pada model klasifikasi.



Gambar 3. Grafik Loss dan Accuracy Hasil Training EfficientNetV2-S

Hasil dari *performance metrics* menggunakan *test set* sebanyak 154 citra menunjukkan bahwa untuk bisa melakukan klasifikasi pada kesegaran dari parameter mata dengan label 'eye-fresh' model klasifikasi yang dihasilkan mampu mencapai *precision* sebesar 90% dan *recall* sebesar 93%. Sementara pada label 'eye-non-fresh' memiliki *precision* sebesar 95% dan *recall* sebesar 93%. Sementara pada klasifikasi kesegaran dari parameter insang, model klasifikasi mampu mencapai *precision* sebesar 94% dan *recall* sebesar 94% untuk label 'gill-fresh', serta *precision* sebesar 90% dan *recall* sebesar 90% untuk label 'gill-non-fresh'. Hal ini menunjukkan bahwa model klasifikasi yang dibangun sudah mampu melakukan klasifikasi kesegaran ikan pada parameter mata dan insang dengan baik. Dari hasil tersebut akurasi model secara keseluruhan dari hasil evaluasi menggunakan *test set* adalah sebesar 93%. Hasil dari *performance metrics* dapat dilihat pada Gambar 4.

	precision	recall	f1-score	support
eye-fresh	0.90	0.93	0.92	29
eye-non-fresh	0.95	0.93	0.94	43
gill-fresh	0.94	0.94	0.94	51
gill-non-fresh	0.90	0.90	0.90	31
accuracy			0.93	154
macro avg	0.92	0.93	0.93	154
weighted avg	0.93	0.93	0.93	154

Gambar 4. *Performance Metrics* Hasil Training EfficientNetV2-S

4. SIMPULAN

Berdasarkan hasil eksperimen dan pengujian, model deteksi menggunakan algoritma YOLOv4-Tiny mampu memberikan *mean average precision* sebesar 93,58% dalam melakukan deteksi ROI mata dan insang dari ikan Gurami. Sementara model klasifikasi kesegaran ikan Gurami menggunakan arsitektur EfficientNetV2-S berhasil diterapkan dengan memanfaatkan metode *transfer learning* dan *optimizer* menggunakan Adam. Tingkat akurasi terbaik yang berhasil dicapai oleh model klasifikasi untuk bisa melakukan prediksi kesegaran ikan Gurami dari ROI mata dan insang dalam penelitian ini adalah sebesar 93%. Hasil tersebut menunjukkan bahwa model yang dihasilkan dari kedua algoritma dapat dipakai untuk melakukan deteksi kesegaran ikan Gurami secara organoleptik dari segi visual.

DAFTAR PUSTAKA

- [1] Kementerian Kelautan dan Perikanan, "Produksi Perikanan," 2021. https://statistik.kkp.go.id/home.php?m=prod_ikan_prov&i=2#panel-footer-kpda (accessed May 11, 2023).
- [2] E. Suprayitno, "Kajian Kesegaran Ikan Di Pasar Tradisional Dan Modern Kota Malang," *Journal of Fisheries and Marine Research*, vol. 4, no. 2, pp. 289–295, Jul. 2020, doi: <http://dx.doi.org/10.21776/ub.jfmr.2020.004.02.13>.
- [3] S. Mitra, Most. N. Khatun, Md. M. H. Prodhan, and Md. A. Khan, "Consumer preference, willingness to pay, and market price of capture and culture fish: Do



- their attributes matter?," *Aquaculture*, vol. 544, p. 737139, Nov. 2021, doi: 10.1016/j.aquaculture.2021.737139.
- [4] Badan Standardisasi Nasional, *Ikan Segar*. Jakarta, 2013. Accessed: May 28, 2023. [Online]. Available: www.bsn.go.id
- [5] J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Machine Learning with Applications*, vol. 6, p. 100134, Dec. 2021, doi: 10.1016/j.mlwa.2021.100134.
- [6] M. Christiawan, L. Willyanto Santoso, and D. Haryadi Setiabudi, "Deteksi Tingkat Kesegaran Ikan Menggunakan Metode Convolutional Neural Network Dengan Parameter Mata dan Warna Insang," Surabaya, 2020.
- [7] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Comput Sci*, vol. 199, pp. 1066–1073, 2022, doi: 10.1016/j.procs.2022.01.135.
- [8] E. Prasetyo, N. Suciati, and C. Fatichah, "Yolov4-tiny with wing convolution layer for detecting fish body part," *Comput Electron Agric*, vol. 198, p. 107023, Jul. 2022, doi: 10.1016/j.compag.2022.107023.
- [9] E. Prasetyo, R. Purbaningtyas, R. D. Adityo, N. Suciati, and C. Fatichah, "Combining MobileNetV1 and Depthwise Separable convolution bottleneck with Expansion for classifying the freshness of fish eyes," *Information Processing in Agriculture*, vol. 9, no. 4, pp. 485–496, Dec. 2022, doi: 10.1016/j.inpa.2022.01.002.
- [10] M. Tan and Q. v. Le, "EfficientNetV2: Smaller Models and Faster Training," Apr. 2021, Accessed: Jan. 16, 2023. [Online]. Available: <https://arxiv.org/abs/2104.00298>
- [11] APJII, "Profil Internet Indonesia 2022," *Asosiasi Penyelenggara Jasa Internet Indonesia (APJII)*, no. June, 2022, Accessed: Jan. 22, 2023. [Online]. Available: <https://apjii.or.id/survei/surveiprofilinternetindonesia2022-21072047>
- [12] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, "Transfer learning: a friendly introduction," *J Big Data*, vol. 9, no. 1, p. 102, Oct. 2022, doi: 10.1186/s40537-022-00652-w.
- [13] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," Dec. 2020, Accessed: Jan. 22, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2011.08036>
- [14] Keras, "Keras API Reference," 2022. <https://keras.io/api/> (accessed May 27, 2023).
- [15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int J Comput Vis*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.
- [16] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *Int J Comput Vis*, vol. 111, no. 1, pp. 98–136, Jan. 2015, doi: 10.1007/s11263-014-0733-5.