

# Segmentasi Buah Apel Menggunakan Framework YOLACT Arsitektur Resnet-101

I Kadek Gunawan, I Putu Agung Bayupati, Kadek Suar Wibawa

Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana

Bukit Jimbaran, Bali, Indonesia, telp. (0361) 701806

E-mail: gunawankadek@student.unud.ac.id, bayupati@unud.ac.id, suar\_wibawa@unud.ac.id

## Abstrak

Apel merupakan salah satu buah yang populer di dunia. Apel pertama kali tumbuh di Kawasan Asia tengah kemudian berkembang luas ke wilayah yang lebih dingin. Apel dapat diolah menjadi minuman kemasan. Produksi pengolahan apel, dibutuhkan proses untuk memilah apel baik berdasarkan kematangan, kondisi buah, ataupun ukuran buah apel sesuai dengan kebutuhan. Pada produksi skala besar hal ini menjadi tantangan karena membutuhkan tenaga kerja yang tidak sedikit. Penelitian ini mengusulkan sistem segmentasi buah apel dengan menggunakan Framework YOLACT dengan arsitektur Resnet-101 yang merupakan salah satu arsitektur *Convolutional Neural Network*. Sistem yang dibuat diharapkan mempermudah pemilahan ataupun penghitungan apel pada manufaktur skala besar. Proses segmentasi menggunakan YOLACT terdiri dari *Feature backbone, Feature Pyramid Network, Protonet, Prediction Head, NMS, cropping dan thresholding*. Tiga jenis apel digunakan dalam penelitian ini yaitu *Fuji, Red Delicious, dan Gala Apple*. mAP rata rata tertinggi diperoleh dengan model YOLACT iterasi 10000, yaitu sebesar 83,12%.

**Kata kunci:** *Apel, Convolutional Neural Network, Resnet-101, Segmentasi, YOLACT*

## Abstract

*Apple is one of the most popular fruit in the world. Apples were first grown in the Central Asian Region and then expanded into colder regions. Apples can be processed into packaged drinks. In an apple fruit processing manufacturer, it takes a good process to sort apples by maturity and condition of the fruit as needed. In large scale production, this becomes a challenge because it requires plenty of labour. This study proposes System that capable of segmenting apples using the YOLACT framework with Resnet-101. The system created is expected to facilitate sorting or counting apples in large-scale manufacturing. The segmentation process using YOLACT consists of Feature backbone, Feature Pyramid Network, Protonet, Prediction Head, NMS, cropping and thresholding. Three types of apples are used in this study, namely Fuji, Red Delicious, and Gala Apple. The highest average mAP is obtained by the YOLACT model iteration 10000, which is 83.12%.*

**Keywords:** *Apple, Convolutional Neural Network, Resnet-101, Segmentasi, YOLACT*

## 1. Introduction

Apel merupakan salah satu buah yang populer di dunia. Apel pertama kali tumbuh di Kawasan Asia tengah kemudian berkembang luas ke wilayah yang lebih dingin[1]. Produksi buah apel di seluruh dunia mencapai 70 juta Ton pada tahun 2018 sampai dengan 2019. Republik Rakyat Cina menjadi produsen apel terbesar didunia dengan jumlah produksi mencapai 33 juta Ton di tahun yang sama[2]. di Indonesia apel mulai dibudidayakan pada tahun 1934 di daerah Malang. Apel memiliki banyak jenis berbeda yang dapat diolah menjadi minuman kemasan. Pada produksi pengolahan apel, dibutuhkan proses untuk memilah apel baik berdasarkan kematangan, kondisi buah, ataupun ukuran buah apel sesuai dengan kebutuhan produksi. Pada produksi skala besar hal ini menjadi tantangan karena membutuhkan tenaga kerja yang tidak sedikit. Sejalan dengan perkembangan teknologi, pekerjaan tersebut dapat dilakukan secara otomatis dengan bantuan komputer.

Beberapa penelitian sebelumnya yang telah mencoba untuk meneliti segmentasi buah apel, yaitu dilakukan oleh Joding et.al melakukan segmentasi buah apel yang terbungkus dengan plastik. Metode yang digunakan yaitu *Contrast Limited Adaptive Histogram Equalization* dan *Otsu thresholding*. Percobaan dilakukan dengan kondisi pencahayaan normal dan kondisi cahaya *backlight*. hasil percobaan pada pencahayaan normal diperoleh *segmentation error*,

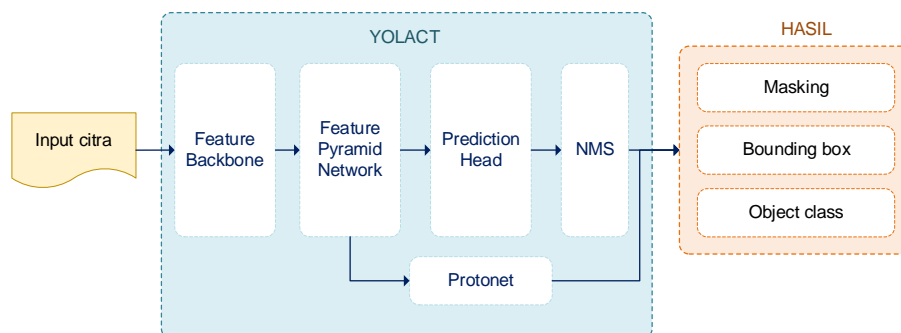
---

*False positive rate* dan *false negative rate* sebesar 12.15%, 8.79% and 11.83%, sedangkan pada kondisi *backlight* diperoleh *segmentation error*, *False positive rate* dan *false negative rate* 19.20%, 17.24% and 17.29% [3]. Penelitian lainnya mengenai segmentasi apel dilakukan oleh Xiaowei Wang et. Al [4], penelitian ini melakukan segmentasi buah apel Fuji yang masih berada pada pohon secara otomatis. Metode yang digunakan, yaitu CNN dengan arsitektur GoogleNet untuk mendapatkan fitur citra apel, lalu *Modified Fuzzy C-means* digunakan untuk melakukan *clustering* objek, terakhir SVM dengan *Gaussian kernel* digunakan untuk melakukan klasifikasi objek. Hasil yang didapatkan dalam penelitian ini yaitu akurasi sebesar 96,5% dan IoU threshold sebesar 76,5%. Penelitian lainnya dilakukan oleh akira Mizushima dan Renfu Lu yang melakukan segmentasi apel dengan menggunakan Linear SVM dan Otsu *thresholding*. Linear SVM digunakan untuk meningkatkan kontras antara objek apel dan background. Hasil yang didapatkan dalam penelitian ini yaitu tingkat *error rate* sebesar 1,42% dari 300 citra data uji [5]. Penelitian mengenai segmentasi apel juga dilakukan oleh Chunlong Zhang, Kunlin Zou dan Yue Pan. Penelitian ini melakukan segmentasi apel yang masih berada di pohonnya dan membandingkan beberapa metode machine learning yang digunakan untuk melakukan segmentasi. Metode yang digunakan yaitu *Grey-Level Co-occurrence Matrix* (GLCM) dan *Principal component analysis* (PCA) dan metode *machine learning*. GLCM digunakan untuk mendapatkan fitur tekstur dari citra. Proses selanjutnya dilakukan *dimensional reduction* dengan menggunakan metode PCA dimana nilai input yang digunakan yaitu nilai dari GLCM, citra awal (RGB), konversi citra HSV dan *greyscale*. Hasil dari metode PCA tersebut selanjutnya digunakan kedalam beberapa metode *machine learning*. Hasil terbaik diperoleh dengan menggunakan metode *Random Forest* dengan akurasi 94%. Sedangkan untuk nilai *segmentation error*, *False positive rate* dan *false negative rate* masing masing sebesar 7%, 13% dan 15% [6].

Penelitian ini mengusulkan pembuatan sistem untuk mampu melakukan segmentasi terhadap buah apel dengan menggunakan Framework YOLACT[7] dengan arsitektur Resnet-101 [8]. Sistem yang dibuat diharapkan mempermudah pemilahan ataupun penghitungan apel pada manufaktur skala besar.

## 2. Research Method / Proposed Method

Penelitian ini menggunakan 3 spesies apel yang berbeda antara lain *Fuji*, *Red Delicious*, dan *Gala Apple*. Citra apel yang digunakan memiliki resolusi 320 x 258 pixel dengan variasi pencahayaan dan posisi buah. Pembuatan sistem segmentasi apel dilakukan menggunakan Framework YOLACT dengan arsitektur Resnet-101. Gambaran umum dari sistem segmentasi apel dapat dilihat pada Gambar 1



**Gambar 1.** Proses Segmentasi YOLACT

Gambar 1 menampilkan proses segmentasi apel dengan menggunakan *Framework* YOLACT. Pada proses YOLACT beberapa tahapan, antara lain *Feature Backbone* untuk mendapatkan *feature map* dari citra *input*, lalu *Feature Pyramid Network* dengan *Prediction Head* untuk melakukan proses pendeteksian lokalisasi kandidat objek, NMS digunakan untuk menggabungkan beberapa kandidat objek yang masih saling berdekatan dan Protonet untuk meng-generate *masking* dari tiap objek. Hasil dari proses YOLACT berupa *masking*, *bounding box* dan kelas dari objek yang terdeteksi didalam sebuah citra.

## 2.1 Tahap Pengumpulan Data

Citra input yang digunakan berupa citra apel dengan 3 spesies berbeda. Terdapat 400 citra pada tiap kelasnya, yang selanjutnya dibagi menjadi 300 data training dan 100 data testing di setiap kelasnya. Data yang akan dilakukan proses training sebelumnya dilakukan proses labelling. Data yang digunakan didapat dari Dataset fruit Recognition Version 1[9].



Gambar 2. Dataset apel

## 2.2 Tahap Labelling Data

Tahap *labelling* data dilakukan pada data *training* dan data *test*. Labelling merupakan proses melakukan anotasi dan penamaan pada tiap objek pada data citra. Proses labelling dilakukan dengan menggunakan tools labelme[10].

## 2.3 Tahap Training Data

Tahap training data dilakukan untuk mempelajari representasi dari data citra yang sudah dilakukan *labelling* sebelumnya. Proses *training data* dilakukan secara berulang berdasarkan jumlah iterasi maupun *epoch*. Proses evaluasi data biasanya dilakukan dalam beberapa iterasi maupun *epoch* untuk mengetahui bagaimana performa dari model. Hasil dari proses *training* disimpan untuk nantinya digunakan dalam proses pengujian.

## 2.4 Tahap Segmentasi dan Klasifikasi

Tahap segmentasi merupakan tahap mempartisi citra digital menjadi beberapa kelompok berdasarkan ciri *pixel* didalam citra tersebut. citra apel yang masuk kedalam input data Tahap segmentasi dan klasifikasi merupakan tahap pengujian untuk dapat memisahkan background dan objek yang terdapat didalam sebuah citra dan melakukan pengenalan dari objek yang disegmentasi tersebut.

## 3. Literature Study

Kajian pustaka memuat materi yang menjadi referensi penelitian. Referensi yang digunakan disesuaikan dengan topik penelitian yang dilakukan. Topik pada penelitian ini adalah segmentasi buah apel. Referensi yang dimuat yakni terkait *deep learning*, augmentasi, *Convolutional Neural Network*, *Fully Convolutional Network*, *Feature Pyramid Network*, dan YOLACT

### 3.1 Deep Learning

*Deep learning* merupakan bagian dari machine learning yang memiliki model komputasi yang tersusun atas beberapa layer pemrosesan untuk mempelajari representasi data secara otomatis. Deep learning dapat mempelajari data yang massif dengan menggunakan back propagation untuk mampu melakukan perubahan parameter internal, dimana parameter internal ini digunakan untuk menghitung representasi di setiap layer dari representasi layer sebelumnya. Setiap Layer tersusun atas komposisi operasi non-linier [11].

### 3.2 Convolutional Neural network

CNN atau Convolutional Neural Network merupakan salah satu metode deep learning yang biasanya digunakan dalam pengolahan data citra untuk melakukan klasifikasi dalam sebuah gambar atau video. CNN didesain untuk mampu secara otomatis mempelajari dari fitur yang ada. CNN memiliki 3 jenis layer yaitu convolution, pooling, dan fully connected layers. Convolution dan pooling layer digunakan untuk melakukan ekstraksi fitur dalam sebuah citra, sedangkan fully connected layer memetakan fitur yang diekstraksi menjadi hasil akhir, seperti

melakukan klasifikasi. Dalam proses CNN, dilakukan proses training yang berguna untuk mengoptimisasi parameter seperti kernel, sehingga meminimalkan perbedaan antara output dan ground truth menggunakan algoritma optimisasi yang disebut back propagation dan gradient descent [12].

### 3.3 Augmentasi

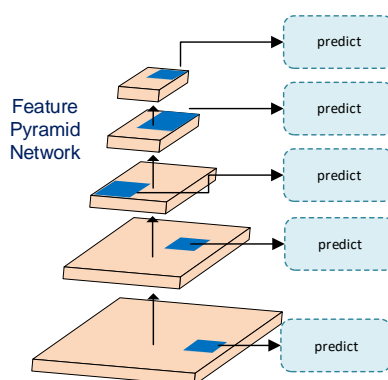
Augmentasi data adalah proses memperbanyak data latih dengan kondisi tertentu. Augmentasi berguna ketika jumlah data training sedikit, dimana metode deep learning memerlukan variasi data yang sangat banyak agar model yang dibuat dapat mengenali dengan baik, selain itu augmentasi bertujuan agar dapat menggeneralisasi gambar yang dideteksi. Terdapat bermacam-macam teknik untuk melakukan augmentasi, seperti melakukan variasi zooming, rotasi, flip dan shearing pada sebuah citra [13].

### 3.4 Fully Convolutional Network

*Fully Convolutional Network* merupakan metode yang digunakan untuk melakukan *instance segmentation* pada sebuah citra. FCN menggunakan *Convolutional Neural Network* untuk mengelompokkan tiap *pixel* pada sebuah citra[14].

### 3.5 Feature Pyramid Network

*Feature pyramid Network* atau FPN merupakan salah satu metode ekstraksi fitur yang menggunakan konsep piramida pada layer network. FPN digunakan untuk melakukan segmentasi objek yang terdapat didalam citra. FPN biasanya bekerja dengan menggunakan nilai beberapa layer feature map yang berbeda dari Fully convolutional Network yang dihubungkan dengan lateral connection. Proses deteksi dilakukan dengan *sliding window* di masing masing layer FPN [15].

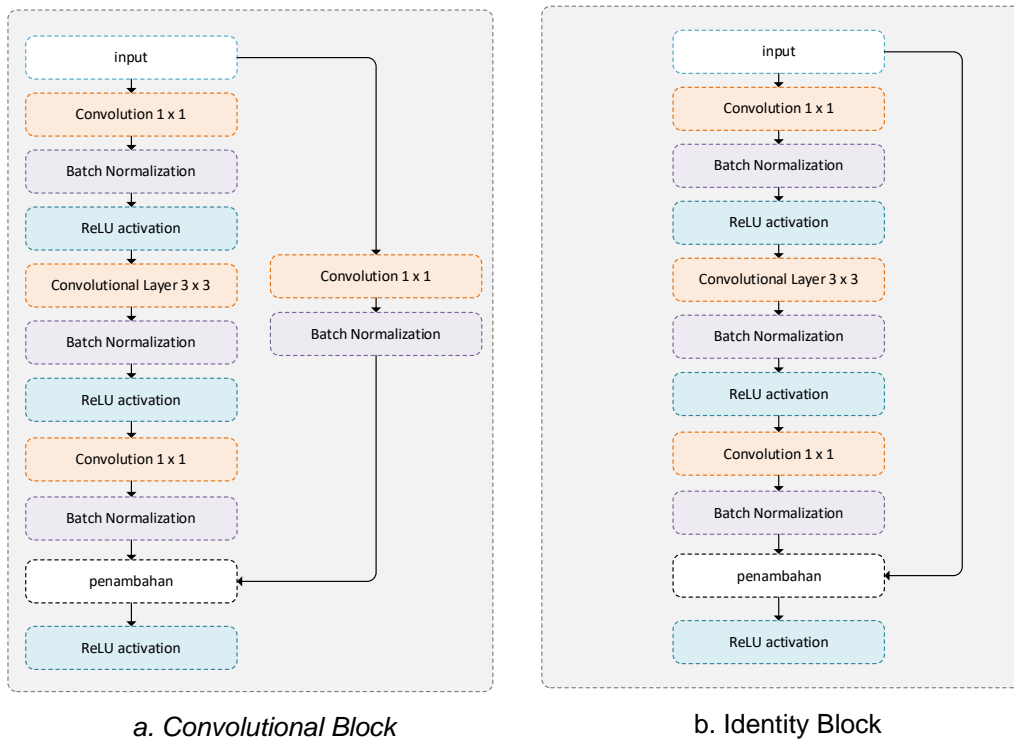


Gambar 3. Feature Pyramid Network

Gambar 3 menampilkan *Feature Pyramid Network* dengan skema *bottom up*. Gambar kotak biru tua yang terdapat pada setiap layer FPN merupakan proses *sliding window* untuk melakukan proses deteksi objek.

### 3.6 Resnet 101

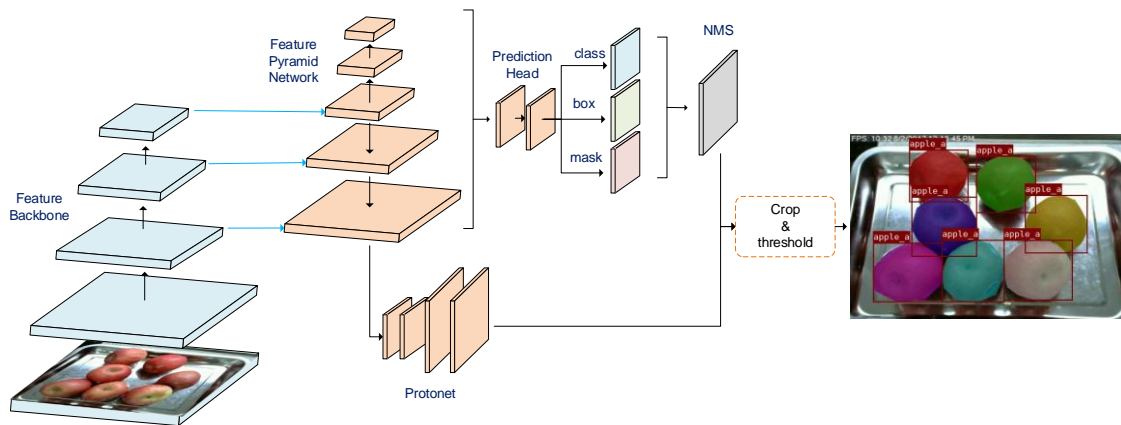
Resnet merupakan salah satu arsitektur *Convolutional Neural Network* yang dikembangkan oleh peneliti Microsoft pada tahun 2015. Resnet dibuat untuk dapat mengatasi *degradation problem* dimana ketika *layer network* yang dalam akan mengalami penurunan akurasi pada proses *training*, bahkan jika dibandingkan dengan arsitektur CNN yang lebih sederhana, padahal dengan jumlah *layer* yang lebih banyak *network* memiliki lebih banyak parameter untuk mampu mempelajari citra tersebut. *Residual Block* digunakan untuk mengatasi permasalahan tersebut. Residual Block merupakan jaringan arsitektur dimana terdapat *skip connection* antara layer. Residual Block terbagi menjadi 2 jenis yaitu *Convolutional Block* dan *Identity Block* [8].



Gambar 4. Residual Block

### 3.7 YOLACT

YOLOACT[7] merupakan *framework* yang digunakan untuk melakukan *instance segmentation* pada sebuah citra atau video. YOLOACT menggunakan metode *deep learning* untuk mampu memisahkan sebuah objek satu dengan lainnya pada sebuah citra dan melakukan identifikasi dari setiap objek yang dideteksi. Framework YOLOACT menyediakan tiga *backbone* yang dapat digunakan yaitu Resnet-101[8], ResNet-50[8], DarkNet-53 dan VGG19[16]. Gambar arsitektur YOLOACT bisa dilihat pada gambar 5



Gambar 5. Arsitektur YOLACT

Proses pengenalan dan segmentasi YOLACT terdiri dari beberapa tahapan. Gambar 4 menampilkan tahapan proses dimulai dari *Feature backbone*, *Feature Pyramid Network*, *Protonet*, *Prediction Head*, *NMS*, *cropping* dan *thresholding*. Proses pada *Feature Backbone*

merupakan *Fully Convolutional Network* dengan beberapa arsitektur CNN seperti Resnet-50, Resnet-101, Darknet 53 dan VGG 19. *Feature Backbone* berfungsi untuk mendapatkan *feature map* dari citra input. *Layer* ke-3 sampai *Layer* ke-5 selanjutnya masuk ke *Feature Pyramid Network* melalui *lateral connection*. Pada *lateral connection* terdapat proses konvolusi untuk mengubah dimensi dari *feature map*. *Feature Pyramid Network* dengan Prediction Head akan mendapatkan kandidat *bounding box*, kelas dan *masking*. Kandidat *bounding box* dan kelas selanjutnya dilakukan proses NMS untuk menentukan *bounding box* yang paling presisi. Pada YOLACT juga terdapat proses Protonet. Protonet berfungsi untuk menghasilkan *masking* yang lebih baik. Hasil *masking Prediction Head* dan Protonet selanjutnya dilakukan penggabungan. *Crop dan threshold* dilakukan pada proses terakhir untuk melakukan *masking* pada citra asli dengan *masking* yang sudah didapatkan sebelumnya menggunakan *Prediction Head* dan Protonet.

### 3.8 COCO Detection Evaluation

COCO detection Evaluation[17] merupakan salah satu cara untuk melakukan evaluasi terhadap performa deteksi dan segmentasi yang dilakukan pada sebuah citra. COCO Detection Evaluation menggunakan nilai dari Intersection over Union (IoU), *precision*, dan *recall* untuk melakukan proses *benchmarking*.



Gambar 6. IoU dengan nilai 0,5

Proses *benchmarking* COCO menggunakan nilai dari IoU. IoU melakukan perhitungan seberapa dekat sebuah *bounding box* dalam memprediksi ground truth dari objek. Gambar 4 memvisualisasikan IoU, dimana kotak hijau merupakan *ground truth*, kotak biru merupakan prediksi *bounding box* sedangkan kotak merah merupakan nilai dari IOU.

$$precision = \frac{TP}{TP+FP} \quad (2.1)$$

Precision digunakan untuk mengukur seberapa akurat prediksi yang dilakukan oleh sistem. Variabel *TP* merupakan total data yang diprediksi sesuai dengan kelas data tersebut dan Variabel *FP* merupakan total data yang salah diprediksi sebagai kelasnya

$$recall = \frac{TP}{TP+FN} \quad (2.2)$$

Recall digunakan untuk mengukur seberapa baik model yang digunakan mendapatkan semua kemungkinan positif dari pengujian yang dilakukan. Variabel *FN* merupakan total data yang diprediksi tidak sesuai kelasnya.

$$p_{interp}(r) = \max_{r' \geq r} p(r') \quad (2.3)$$

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}) \quad (2.4)$$

$$mAP = \sum_{i=1}^K AP_i \quad (2.5)$$

*mAP* atau Mean Average precision menghitung nilai rata rata *AP* dari masing masing kelas. Nilai *AP* dihitung berdasarkan nilai *p<sub>interp</sub>* antara *precision* dan *recall*. *K* merupakan jumlah kelas data yang ada dan  $r_1, r_2, r_3, \dots, r_n$  merupakan hasil dari proses *p<sub>interp</sub>*. Pada COCO

detection Evaluation nilai *mAP* juga dihitung berdasarkan nilai IoU. *mAP*<sup>50</sup> memiliki arti *mAP* dengan nilai IoU sebesar 0,5.

#### 4. Result and Discussion

Bagian hasil dan pembahasan membahas mengenai proses training dan proses pengujian yang dilakukan. Proses training terdiri dari tahap labelling, augmentasi data dan hasil training dengan parameter epoch, sedangkan proses pengujian menampilkan proses segmentasi yang dilakukan dengan Framework YOLACT.

##### 4.1 Proses labelling data

Proses labelling data dilakukan untuk menandai lokasi setiap objek yang didalam satu citra dan memberi penamaan dari lokasi tersebut sesuai dengan kelas yang ada. Proses labelling dilakukan menggunakan *tool/Labelme*.



Gambar 7. Proses *labelling* data dengan Labelme

##### 4.2 Proses Augmentasi

Augmentasi dilakukan untuk memperbanyak ciri pada prose pelatihan. Proses augmentasi dilakukan sesuai dengan proses bawaan yang terdapat pada Framework Yolact. Proses augmentasi pada *Framework* YOLACT menggunakan augmentasi yang sama dengan SSD[18]. Gambar 8 menampilkan citra hasil augmentasi.



Gambar 8. Citra hasil Augmentasi

##### 4.3 Hasil Training

Pada proses training yang dilakukan menggunakan parameter jumlah iterasi. Hasil proses training dengan evaluasi menggunakan *COCO Detection Evaluation* dapat dilihat pada tabel 1.



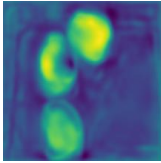


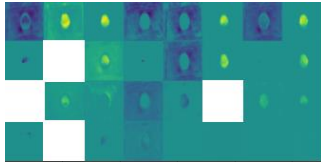
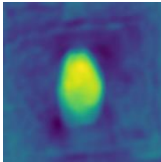
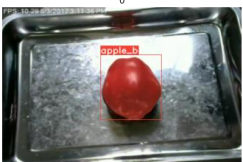

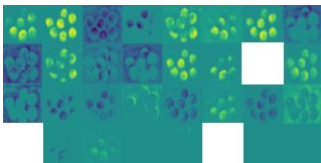
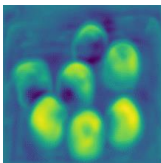
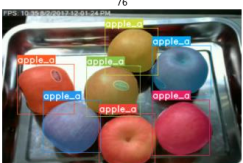
Tabel 1. Hasil Proses *Training*

No	Jumlah Iterasi	mAP	mAP <sup>50</sup>	mAP <sup>75</sup>	mAP <sup>90</sup>
1	2500	69,22	93,03	89,92	10,25
2	5000	80,00	98,28	95,89	38,55
3	7500	82,87	98,89	97,07	52,34
4	10000	83,12	98,91	97,44	53,77

#### 4.4 Hasil Segmentasi

Hasil dari proses segmentasi menampilkan citra apel yang diberi *bounding box*, *masking* beserta prediksi kelas apel. Hasil dari segmentasi dapat dilihat pada Tabel 2.

**Tabel 2.** Hasil Segmentasi

Citra Input	prototypes	ProtoNet	Hasil
			
			
			

Tabel 2 merupakan hasil proses segmentasi yang dilakukan. Citra input merupakan citra yang digunakan untuk proses pengujian segmentasi. Prototypes merupakan visualisasi dari layer Protonet yang terdiri dari 32 layer sebelum akhirnya dilakukan proses penggabungan. ProtoNet merupakan hasil dari penggabungan prototypes. Dan hasil merupakan citra asli yang telah diberi *bounding box*, *masking* dan kelas dari objek yang terdeteksi.

#### 4.5 Hasil Uji Coba.

Proses pengujian dilakukan dengan menggunakan 50 data uji dari masing masing kelas. Terdapat 150 data pengujian dengan kondisi pencahayaan, jumlah apel dan jenis apel yang berbeda. Evaluasi pengujian yang dilakukan menggunakan *COCO Detection Evaluation*. Hasil pengujian dapat dilihat pada tabel 3.

**Tabel 3.** Hasil Proses Pengujian

No	Jumlah Iterasi	mAP	mAP <sup>50</sup>	mAP <sup>75</sup>	mAP <sup>90</sup>
1	2500	65,83	89,00	86,85	8,40
2	5000	76,34	95,67	92,66	29,97
3	7500	80,34	98,42	93,94	44,07
4	10000	80,77	98,42	92,40	47,12

#### 5. Conclusion

Sistem segmentasi buah apel menggunakan *Framework* YOLACT dengan Arsitektur Resnet-101 berfungsi untuk mempermudah pemilahan apel pada proses produksi skala besar. *Framework* YOLACT digunakan untuk melakukan segmentasi dan klasifikasi buah apel. Terdapat 3 jenis buah apel digunakan dalam penelitian ini yaitu apel *Fuji*, *Red Delicious*, dan *Gala Apple*. Proses YOLACT terdiri dari beberapa tahapan, yaitu *Feature backbone*, *Feature Pyramid Network*, *Protonet*, *Prediction Head*, *NMS*, *cropping* dan *thresholding*. Pada proses pengujian sistem, 4 model YOLACT yang berbeda digunakan yaitu, iterasi 2500, 5000, 7500



dan 10000 terhadap 150 data uji. mAP rata rata tertinggi diperoleh dengan model YOLACT iterasi 10000, yaitu sebesar 83,12%.

#### Daftar Pustaka

- [1] Suhariyono, "Sejarah Perkembangan Apel di Indonesia." <http://balitjestro.litbang.pertanian.go.id/sejarah-perkembangan-apel-di-indonesia/> (accessed Dec. 09, 2020).
  - [2] "Fresh Apples, Grapes, and Pears: World Markets and Trade." United States Department of Agriculture, Dec. 2019, [Online]. Available: <https://downloads.usda.library.cornell.edu/usda-esmis/files/1z40ks800/pc289z933/3484zz05r/fruit.pdf>.
  - [3] J. Lv, F. Wang, L. Xu, Z. Ma, and B. Yang, "A segmentation method of bagged green apple image," *Scientia Horticulturae*, vol. 246, pp. 411–417, 2019, doi: <https://doi.org/10.1016/j.scienta.2018.11.030>.
  - [4] X. Wang, S. Yin, K. Sun, H. Li, J. Liu, and S. Karim, "GKFC-CNN: Modified Gaussian Kernel Fuzzy C-means and Convolutional Neural Network for Apple Segmentation and Recognition," *Journal of Applied Science and Engineering*, vol. 23, no. 3, pp. 555–561, Sep. 2020, doi: 10.6180/jase.202009\_23(3).0020.
  - [5] A. Mizushima and R. Lu, "An image segmentation method for apple sorting and grading using support vector machine and Otsu's method," *Computers and Electronics in Agriculture*, vol. 94, pp. 29–37, 2013, doi: <https://doi.org/10.1016/j.compag.2013.02.009>.
  - [6] C. Zhang, K. Zou, and Y. Pan, "A method of apple image segmentation based on color-texture fusion feature and machine learning," *Agronomy*, vol. 10, no. 7, p. 972, 2020.
  - [7] D. Bolya, C. Zhou, F. Xiao, and Y. Lee, *YOLACT: Real-time Instance Segmentation*. 2019.
  - [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016, pp. 770–778.
  - [9] C. Gorgolewski, "Fruit Recognition Dataset Version 1." kaggle.com, 2019, [Online]. Available: <https://www.kaggle.com/chrisfilo/fruit-recognition>.
  - [10] K. Wada, *labelme: Image Polygonal Annotation with Python*. 2016.
  - [11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
  - [12] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.
  - [13] Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, "Differential Data Augmentation Techniques for Medical Imaging Classification Tasks," *AMIA Annu Symp Proc*, vol. 2017, pp. 979–984, 2018.
  - [14] J. L. Evan Shelhamer, "Fully Convolutional Networks for Semantic Segmentation," *PAMI*, 2016.
  - [15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
  - [16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
  - [17] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, Cham, 2014, pp. 740–755.
  - [18] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," *Lecture Notes in Computer Science*, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0\_2.
-