JURNAL ILMIAH FIFO **P-ISSN 2085-4315 / E-ISSN 2502-8332**
DOI: http://dx.doi.org/10.22441/fifo.2022.v14i1.005

Deteksi Objek Daun Semanggi Secara Real Time Menggunakan *CNN-Single Shot Multibox Detector* (SSD)

Ida Astuti*[1], Winda Widya Ariestya[2], Bambang Solehudin[3]

[1,2,3]*Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma*
Jl. Margonda Raya No. 100 Depok, Jawa Barat
*[1]astuti@staff.gunadarma.ac.id, [2]winda_widya@staff.gunadarma.ac.id,
[3]bamsolwibu@gmail.com
*) Korespondensi author

(***received*:** 13-09-21, ***revised*:** 21-12-21, ***accepted*:** 13-01-22)

Abstract

Many medicinal plants are now found as plants that provide benefits in disease prevention and alternative treatments. One of them is clover leaf, which has medicinal properties, but little is known about its shape. This study aims to detect cloverleaf objects in real-time using webcam camera-based image input, followed by deep learning using the Tensorflow object detection framework in the feature extraction process and Convolutional Neural Network (CNN)-Single Shot MultiBox Detector (SSD) as an image processing method. The research method used included several stages, including data acquisition, data preposition, training, model formation, and testing. Based on the test results, the cloverleaf object detection model using SSD can perform well, with the best results in the comparison of the ratio of training data and test data of 80:20 producing a precision value of 80%, recall 100%, and accuracy 86.6 %.
Keyword: object detection, semanggi leaf,real time, tensorflow, SSD

Abstrak

Tumbuhan obat saat ini banyak ditemui sebagai tumbuhan yang memberikan manfaat dalam pencegahan penyakit maupun sebagai alternatif pengobatan. Salah satunya adalah daun semanggi yang memiliki khasiat sebagai tumbuhan obat, namun masyarakat belum banyak mengenal bentuk daun tersebut. Penelitian ini bertujuan untuk melakukan deteksi objek daun semanggi yang dilakukan secara real-time melalui input gambar berbasis kamera webcam, selanjutnya deep learning menggunakan framework Tensorflow object detection digunakan pada proses ekstraksi fitur dan Convolutional Neural Network (CNN)-Single Shot MultiBox Detector (SSD) sebagai metode dalam pengolahan citra. Beberapa tahapan yang dilakukan dalam metode penelitian yang digunakan yaitu akuisisi data, preposisi data, proses training dan pembentukan model serta pengujian. Model deteksi objek daun semanggi menggunakan SSD dapat berjalan dengan baik, hal ini berdasarkan hasil pengujian dengan hasil terbaik pada perbandingan rasio data latih dan data uji sebesar 80:20 menghasilkan nilai precision 80 %, recall 100 % dan akurasi 86,6 %.
***Keyword:** deteksi objek, daun semanggi, real time, tensorflow, SSD*

I. Pendahuluan

Indonesia merupakan negara yang kaya akan sumber daya hayati, salah satunya adalah manfaat tumbuhan sebagai obat. Sebagian dari masyarakat Indonesia masih mempertahankan penggunaan tumbuhan sebagai pengobatan alternatif. Tumbuhan obat adalah tumbuhan yang memiliki fungsi dan berkhasiat sebagai obat dan dipergunakan untuk penyembuhan maupun mencegah berbagai penyakit [1]. Daun semanggi merupakan salah satu tumbuhan yang masih digunakan untuk obat dan tujuan kesehatan.

Daun semanggi merupakan salah satu tumbuhan yang memiliki banyak manfaat diantaranya mengobati infeksi saluran kencing, amandel, batuk, dsb [2]. Masyarakat umum terkadang masih banyak yang belum mengetahui mengenai berbagai macam daun termasuk daun semanggi, sehingga diperlukan suatu sistem yang dapat mengenali daun semanggi.

Pekerjaan manusia sangat terbantu dengan memanfaatkan perkembangan teknologi yang terjadi, *Artificial Intelligence* (AI) atau kecerdasan buatan merupakan salah satu bidang penelitian yang sampai saat ini masih terus berkembang [3]. *Object Detection* merupakan suatu cabang dari AI yang melibatkan proses seperti mengidentifikasi objek melalui gambar, video, maupun sebuah *webcam* [4]. Ciri dari deteksi objek itu sendiri yaitu adanya *bounding box* pada objek yang akan dideteksi dengan persentase kemungkinan suatu objek [5].

Metode yang banyak digunakan untuk menyelesaikan permasalahan yang berkaitan dengan *object detection* dan *image classification* yaitu metode *Convolutional Neural Network* (CNN) [6], kerena memiliki tingkat akurasi yang relatif tinggi dan memiliki hasil yang signifikan dalam pengenalan citra [7]. Terdapat *framework* yang memudahkan dalam penerapan implementasi proses deteksi objek dan CNN yaitu *framework Tensorflow Object Detection API. Framework* tersebut dapat digunakan untuk mengembangkan, melatih, dan menggunakan model deteksi objek [8]. *Framework* ini telah menyediakan beberapa model *trained* yang bisa digunakan, salah satu nya adalah model *Single Shot Multibox Detector* (SSD) [9].

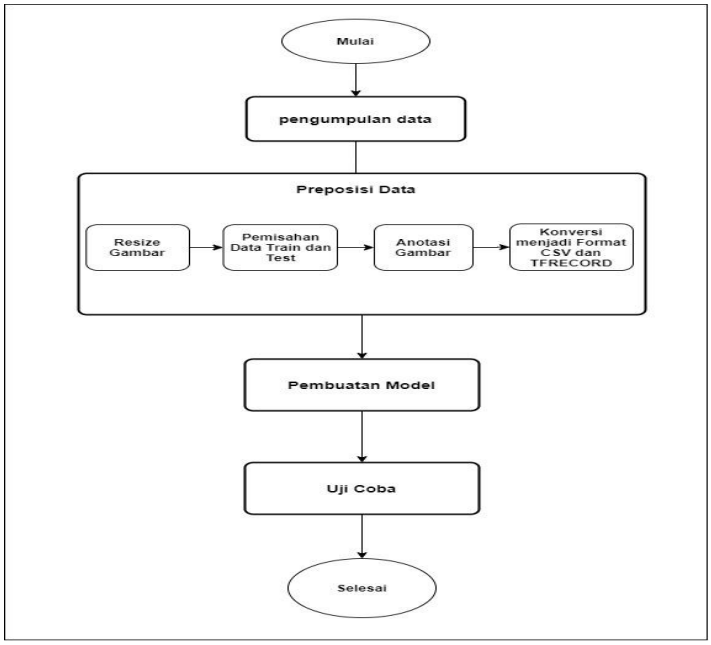
Salah satu metode CNN yang menerapkan *bounding boxes* untuk memperkirakan lokalisasi objek yang dideteksi adalah *Single Shot MultiBox Detector* (SSD). SSD memiliki beberapa kelebihan diantaranya kecepatan dan keakuratan dalam mengenali objek jika dibandingkan dengan model lain seperti *YOLO* dan *Fast R-CNN* [10].

Berikut beberapa penelitian terdahulu yang pernah dilakukan sebelumnya berkaitan dengan pendeteksian objek. Penelitian mengenai "Deteksi objek pada video dengan memanfaatkan metode CNN dan Tensorflow" di tahun 2018, dihasilkan tingkat akurasi berkisar antara 70%-99%. Penelitian mengenai "Implementasi SSD (*Single Shot Multibox Detector*) untuk Pengenalan Wajah" oleh Andrianto Sukusvieri (2020) dihasilkan tingkat akurasi berkisar 88% hingga 100% [11]. Penelitian "Pemanfaatan OpenCV pada deteksi jenis kendaraan di Jalan" di tahun 2017 pada kondisi jalan sepi dihasilkan akurasi 77.8%, 47.5% pada kondisi jalan normal dan 28.2% pada kondisi jalan padat [12]. Penelitian lain yang menghasilkan tingkat akurasi tinggi berkisar antara 70-99% yaitu mengenai "Pemanfaatan CNN dan Tensorflow untuk mendeteksi tanda nomor kendaraan bermotor pada media streaming" pada tahun 2018 [11].

Penelitian ini dilakukan deteksi objek daun semanggi secara realtime dengan mengimplementasikan metode CNN-SSD.

II. Metodologi Penelitian

Pada penelitian ini, beberapa tahapan metodologi penelitian yang digunakan tampak pada Gambar 1.



Gambar 1. Metodologi Penelitian

DOI: http://dx.doi.org/10.22441/fifo.2022.v14i1.005

4 tahapan dalam metode penelitian ini, yaitu:
1. Pengumpulan data
2. Preposisi data
3. Pembuatan model
4. Tahap uji coba

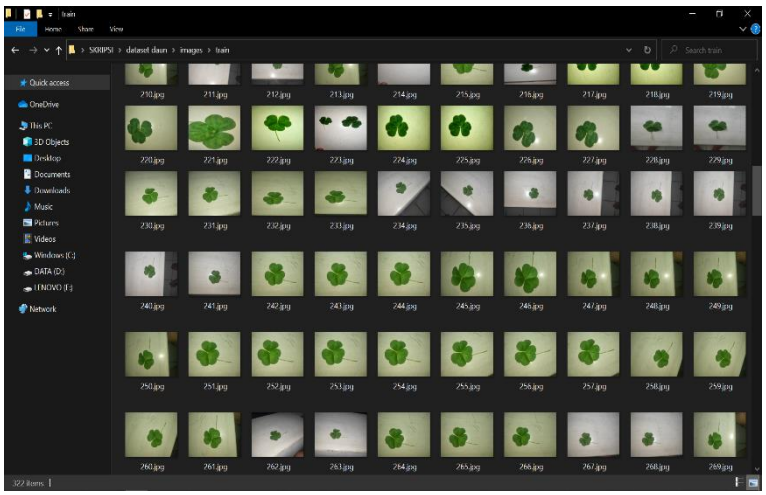
Pengumpulan Data

Prosen pengumpulan data dilakukan pada tahap ini dengan melakukan pencarian citra daun semanggi dan objek daun lain yang terlihat mirip sebanyak 180 buah.

Ciri Daun Semanggi memiliki memiliki tiga helai daun dengan panjang sekitar 2,5 cm dan lebar sekitar 2,3 cm. Daun semanggi mempunyai tekstur tipis dan juga lembut saat diraba. Daunnya berbentuk seperti baji oval telur dengan warna antara hijau muda hingga hijau gelap seperti pada Gambar 2 [2].

Pencarian data image daun semanggi yang didapat melalui media kamera dan internet akan menghasilkan file gambar berekstensi JPG dan PNG. Pengumpulan data melalui media kamera, diambil secara langsung dengan menghasilkan format gambar berupa JPG, sedangkan melalui media internet diambil data melalui *google image*, dan menghasilkan format gambar berupa file JPG dan PNG. Pengumpulan data dapat dilihat pada Gambar 3.

Gambar 2. Daun Semanggi [2]



Gambar 3. Pengumpulan Data

Hasil pengumpulan data terdiri dari 180 *image* daun, dimana 100 merupakan daun semanggi dan 80 adalah daun lain. Komposisi data pada penelitian dapat dilihat pada Tabel 1.

Tabel 1. Komposisi data

No	Jenis Daun	Jumlah
1	Daun Semanggi	100
2	Daun Pepaya	25
3	Daun Singkong	26
4	Daun Jambu	7
5	Daun Kelor	5
6.	Daun Sirih	5
7.	Daun Cincau	6
8.	Daun Mangkokan	6

Preposisi Data

Pada tahap ini setelah data dikumpulkan dilakukan preposisi data. Preposisi data adalah tahapan melakukan persiapan terhadap data sebelum dilakukan pengolahan. Preposisi data pada penelitian ini terbagi menjadi 4, yaitu *resize* gambar, pemisahan data *train* dan *test*, anotasi gambar, dan koversi menjadi format CSV dan TFRecord.

1. *Resize* gambar
   Resize atau mengubah ukuran data dilakukan karena hampir semua gambar mempunyai resolusi yang sangat besar. Dalam penelitian ini seluruh gambar diubah menjadi 800x600.
2. Pemisahan data uji dan latih.
   Pada tahap ini setelah semua data data dilakukan *resize*, data akan dipisah menjadi 2 bagian yaitu data uji dan data latih. Data akan dibagi berdasarkan rasio tertentu.
3. Anotasi Gambar
   Tahap anotasi data gambar menggunakan *software labeling*, dengan cara memberi label pada masing-masing gambar yang ada pada bagian data *train* dan *test*. Pada gambar daun semanggi diberi label "daun semanggi" sedangkan untuk objek lain yang sekilas mirip diberi label 0. Setiap selesai melakukan Anotasi di setiap jenis daun maka format file penyimpanannya berekstensi XML. Proses Anotasi gambar terdapat pada Gambar 4.

Gambar 4. Pemberian Anotasi pada Gambar

4. Konversi menjadi format CSV dan TFRecord
   Data yang tersimpan setelah diberikan anotasi akan berekstensi .xml kemudian dikonversi ke format CSV. Contoh Data CSV dapat dilihat pada Tabel 2. Data CSV merupakan suatu file yang didalamnya terdapat kumpulan data yang sudah dianotasi sebelumnya menjadi satu dalam bentuk tabel. Isi dari kolom pada tabel CSV adalah, *filename* yang merupakan nama file dari gambar, *width* merupakan lebar dari gambar, *height* merupakan tinggi dari gambar, *class* merupakan nama objek pada gambar, *x_min* yaitu nilai jarak kordinat *x-min* pada letak objek yang telah dianotasi sebelumnya, *y_min* yaitu nilai jarak kordinat *y-mi*n pada letak objek yang telah dianotasi sebelumnya, *x_max* yaitu nilai jarak kordinat *x-max* pada letak objek yang telah dianotasi sebelumnya, dan *y_max* yaitu nilai jarak kordinat *y-max* pada letak objek yang telah dianotasi sebelumnya.

Tabel 2. Data CSV

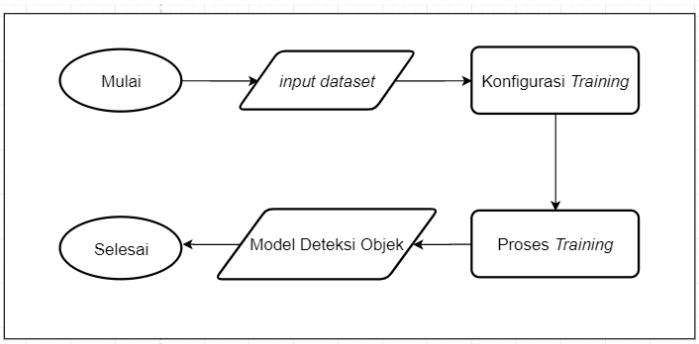
filename,width,height,class,xmin,ymin,xmax,ymax
322.jpg,800,600,daun semanggi,182,154,511, 520
323.jpg,800,600,daun semanggi,325,302,516,489
324.jpg,800,600,daun semanggi,185,154,512,519
325.jpg,800,600,daun semanggi,101,211,371,473
326.jpg,800,600,daun semanggi,327,343,433,440
326.jpg,800,600,daun semanggi,498,271,634,402
327.jpg,800,600,daunsemanggi,7,15,639,385
328.jpg,800,600,daun semanggi,8,13,642,378
329.jpg,800,600,daun semanggi,58,22,524,299
33.jpg,800,600,daun pepaya,223,58,643,456
330.jpg,800,600,daun semanggi,47,24,511,299
331.jpg,800,600,daun semanggi,492,299,686,400
332.jpg,800,600,daun semanggi,376,30,527,426
333.jpg,800,600,daun semanggi,332,113,528,239

Setelah data di ekstrak ke dalam data CSV, maka data akan dikonversi lagi untuk menghasilkan data biner TF Record dengan menjalankan skrip *python generate* TF Record. Setelah dijalankan, akan menghasilkan berkas test.record dan train.record yang dapat digunakan untuk input *dataset* pada saat proses *training.*

Pembuatan Model

Gambar 5 merupakan alur pembuatan model. Tampak bahwa proses pembuatan model terdiri dari 4 tahap yaitu :

1. Input dataset
2. Konfigurasi *training*
3. Proses *training*
4. Model Deteksi Objek



Gambar 5. Proses Pembuatan Model

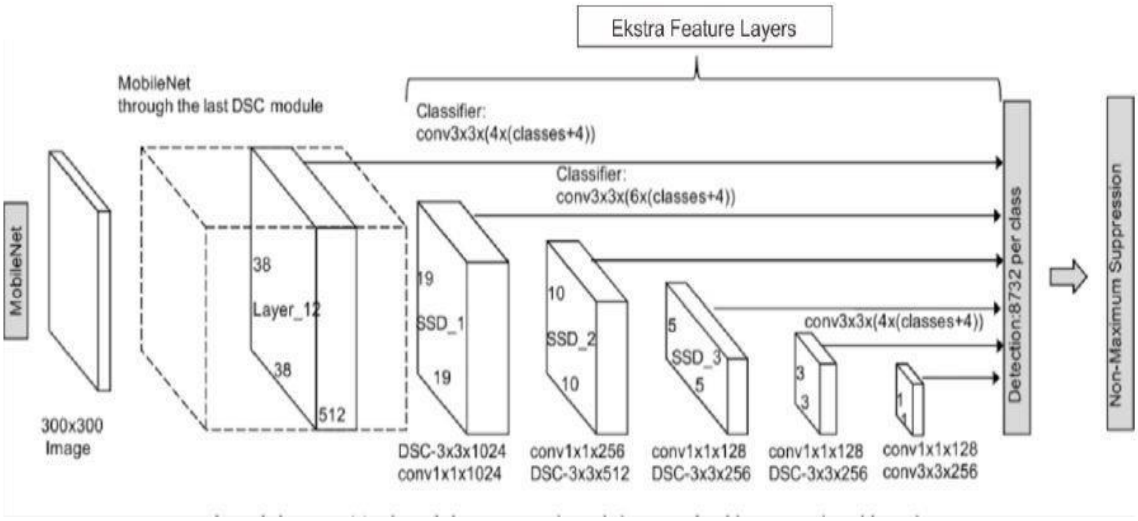
Berikut ini merupakan penjelasan dari tahap pembuatan model :

1. *Input Dataset*
   Tahapan ini merupakan tahapan awal dalam pembuatan model yaitu menyiapkan dataset untuk input pada saat tahap proses *training*. Dataset melalui beberapa ekstraksi yang diawali dari gambar yang kemudian diekstrak ke format CSV dan TFRecord dengan hasil akhirnya berupa train.record, dan test.record yang merupakan input dataset untuk proses training

DOI: http://dx.doi.org/10.22441/fifo.2022.v14i1.005

2. Konfigurasi *Training*
   Konfigurasi yang dilakukan adalah membuat sebuah *label map*, dan juga membuat beberapa perubahan pada berkas ssd_mobilenet_v1_coco. Berkas *label map* berfungsi untuk memberikan prediksi pada objek yang akan dideteksi.
3. Proses *Training*
   Pada proses *Training* menggunakan *framework Tensorflow* [13] *object detection* dan pengolahan citra dengan metode CNN-SSD.

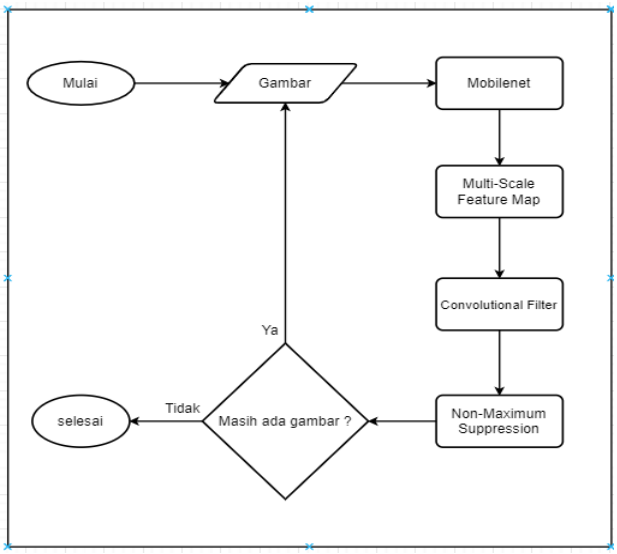
SSD memiliki arsitektur yang dapat dilihat pada Gambar 6. Pada awalnya SSD akan mengubah ukuran pada tiap gambar di dataset menjadi lebih kecil yaitu sebesar 300x300, lalu gambar tersebut akan di konvolusi pada layer *mobilenet. Mobilenet* digunakan sebagai *feature ekstraktor*, setelah itu 5 layer berikutnya merupakan layer utama dari SSD, pada layer – layer tersebut SSD dilengkapi dengan *Multi-Scale Feature Map* yang digunakan untuk menambah keakuratan pada pendeteksian. Pada tahap eliminasi *default box*, SSD menggunakan metode *Non-Maximum Suppression*. Secara total, SSD membuat 8732 prediksi dan menghasilkan akurasi kecepatan yang cukup tinggi yaitu 59 FPS [14].



Gambar 6. Arsitektur SSD

Pada Gambar 7, tampak alur dari proses *training* yaitu:

- Gambar yang ada di *dataset* yang sudah dianotasi sebelumnya akan dilakukan proses konvolusi pada *network* MobileNet. MobileNet juga berfungsi sebagai *feature Ekstraktor* untuk SSD.
- Gambar yang sudah masuk, akan dibuat beberapa *feature map* dengan skala yang berbeda – beda.
- Selanjutnya *Convolutional Filter* akan membuat prediksi beberapa *default box* (lokasi Objek) dan klasifikasi.
- Menggunakan metode *Non-Maximum Suppression*, dilakukan eliminasi *defaultbox* dengan nilai *jaccard* yang lebih kecil, hingga hanya tersisa 1 *default box*, dari masing – masing objek yang terlihat [15].
- Proses diulangi terus menerus hingga gambar yang ada pada *dataset* habis atau sudah mencapai *step* akhir.



Gambar 7. Alur Proses *Training*

4. Model Deteksi Objek
   Pada saat melakukan proses *training*, akan memakan waktu dan selama proses berlangsung, program akan menyimpan hasil berupa berkas .ckpt yaitu *checkpoint* yang disimpan setiap beberapa menit ke folder *training*. Berkas *checkpoint* terakhir akan di ekspor menjadi model baru berupa file *protobuff* yang akan digunakan sebagai acuan model untuk mendeteksi objek.

Tahap Uji Coba

Tahap ini akan dilakukan pengujian pada mekanisme *learning* dengan menggunakan perbandingan rasio data latih dan uji. Selain itu juga dilakukan pengujian terhadap model yang dihasilkan dari hasil *learning* dan dievaluasi dengan *confussion matrix.*

III. Hasil dan Pembahasan

Proses *training* dilakukan dengan melakukan skenario perbandingan antara data *train* dan data *test* yang tampak pada Tabel 3.

Tabel 3. Perbandingan data latih dan uji

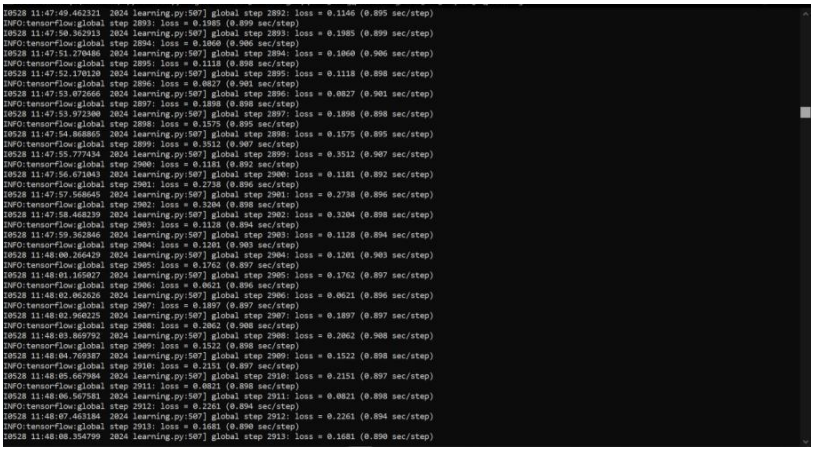
Rasio Data	*Jumlah*	*Data Train*	*Data Test*
60:40	*180*	*108*	*72*
70:30	*180*	*126*	*54*
80:20	*180*	*144*	*36*

Training atau Pelatihan dilakukan pada komputer dengan spesifikasi *processor intel core i7*. Pada proses *training* menggunakan library *SSD-mobilenet* yang sudah tersedia pada *tensorflow* dengan mengatur konfigurasi yang dilakukan merujuk pada penelitian [11]. Berkas train.record dan test.record berisi informasi mengenai dataset gambar beserta anotasinya dan labelmap.pbtx mendefinisikan nama daun semanggi yang dirubah menjadi id agar mudah dikenali oleh komputer. Model akan melakukan resize gambar latih menjadi lebih kecil dengan ukuran 300 x 300. Feature ekstraktor yang digunakan adalah *MobileNet*, kemudian ukuran *batch* yang digunakan adalah 3. Ditentukan juga path menuju *fine tune checkpoint* yaitu berkas model.ckpt.

Data latih yang digunakan berjumlah 180 gambar dengan 100 gambar daun semanggi, dan 80 gambar objek lain yang terlihat mirip yang diberi label 0 saat anotasi gambar. Proses pelatihan dapat dijalankan menggunakan script train.py yang telah disediakan pada direktori tensorflow.

Pelatihan data akan menghasilkan nilai *loss* dari setiap langkahnya. *Loss* adalah angka yang menunjukan seberapa buruk prediksi suatu model. Nilai *loss* adalah 0 menunjukan jika prediksi model sempurna, jika tidak,

nilai nya akan lebih besar dan optimalnya nilai *loss* setidaknya harus dibawah 1. Gambar 8 adalah contoh dari proses *training* yang dilakukan dimana setiap iterasi menghasilkan nilai *loss*.



Gambar 8 Tampilan Proses *Training*

Mendapatkan Model Hasil Prediksi

Pada tahap ini akan diuji dari setiap model yang diperoleh dari hasil *training*, seberapa tinggi tingkat keberhasilan atau akurasi dari hasil deteksi objek yang dilakukan oleh program. Pengujian dilakukan terhadap gambar yang terekam oleh kamera *webcam*. Sebuah objek yang dikenali oleh program akan ditandai dengan munculnya *bounding box* yang mencakup objek yang terlihat.

Gambar 9. Sistem berhasil mendeteksi objek daun semanggi

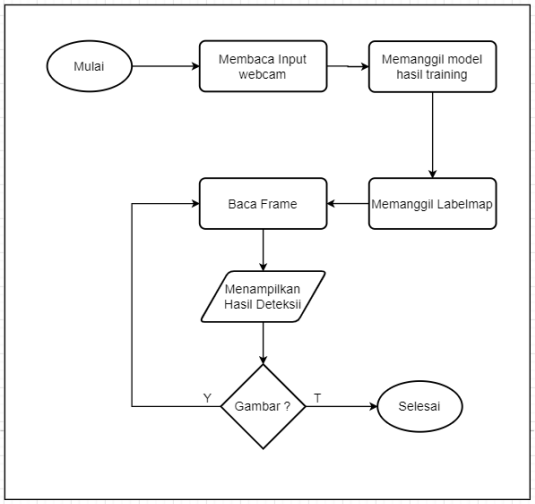

Gambar 10. Sistem berhasil tidak mendeteksi selain daun semanggi

Pada Gambar 9 adalah contoh pengujian bahwa model mampu mendeteksi objek daun semanggi dengan menampilkan *bounding box* pada objek daun semanggi tersebut dengan nilai akurasi di atas 90 %. Pada Gambar 10 adalah contoh pengujian bahwa model mampu tidak mendeteksi objek selain daun semanggi dengan tidak menampilkan *bounding box* pada objek daun tersebut.

Pada hasil uji coba, jika mendapatkan akurasi yang tidak bagus maka akan dilakukan *training* ulang dengan mengubah perbandingan rasio tertentu pada data *train* dan data *test* hingga hasil uji coba mendapatkan akurasi yang terbaik.

Pengujian Implementasi Model Deteksi Objek

Untuk melakukan implementasi dari model yang sudah didapat dari hasil *training* dilakukan secara real time dengan memasukkan input gambar melalui Webcam, dan selanjutnya system akan melakukan prediksi terhadap objek tersebut.



Gambar 11. Alur Implementasi Model

Pada Gambar 11, tampak alur proses dalam fungsi untuk melakukan pendeteksian objek menggunakan model yang sudah melalui proses *training* sebelumnya:

1. Input dari *webcam* akan digunakan untuk mengidentifikasi objek berupa daun semanggi. *Webcam* yang digunakan hanya terdapat pada pc atau laptop.
2. Selanjutnya sistem akan membaca input untuk kemudian diproses untuk dikenali subjek pada *webcam.*
3. Memanggil hasil dari proses pelatihan model yang sudah dikonversi menjadi *file* protobuf, dan memanggil *Label Map* untuk mendapatkan hasil dari prediksi aktivitas.
4. Sistem akan membaca setiap *frame* yang terekam *webcam*, kemudian menampilkan hasil prediksi menggunakan model yang sudah di *training* pada tiap *frame*.
5. Proses membaca *frame* diatas akan dilakukan berulang – ulang, hingga tidak ada lagi *frame* untuk dibaca.

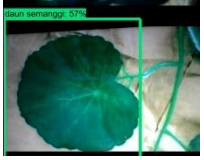
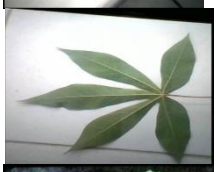
Selanjutnya dilakukan perhitungan *Confusion Matrix* untuk mengevaluasi performance algoritma dari *Machine Learning* (ML). *Confusion Matrix* merepresentasikan prediksi dan kondisi sebenarnya dari data yang dihasilkan oleh algoritma ML

Dalam mengukur performance pada *Confusion Matrix*, terdapat beberapa performance *metrics negative* yang umum dan sering digunakan yaitu: *accuracy*, *precision*, dan *recall* [16].

- *Accuracy*, merupakan rasio prediksi benar (positif dan negative) dengan keseluruhan data.
- *Precission*, merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif
- *Recall* (Sensitifitas), merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif dan salah negative.

Skenario pengujian implementasi model deteksi objek ini dilakukan pada 15 gambar yang terdiri dari 8 gambar daun semanggi dan 7 gambar daun lainnya yang sekilas terlihat mirip. Pengujian dilakukan dari model yang dihasilkan dari hasil *training* dengan menggunakan dengan rasio data *train* dan data *test*, 60:40, 70 :30 dan 80 :20. Dari beberapa pengujian implementasi model deteksi objek yang dilakukan, diperoleh model yang paling baik adalah dengan komposisi data *train* dan data *test* 80:20. Tabel 4 adalah hasil pengujian dari implementasi model yang paling optimal menggunakan 15 gambar. Pada Tabel 4 terlihat bahwa model mampu mendeteksi 8 gambar dengan hasil *True Positive* (TP) dan 5 gambar daun lain yang tidak terdeteksi dengan hasil *True Negative* (TN). Selain itu terdapat 2 gambar yang salah dideteksi oleh model dengan hasil *False Positive* (FP).

Tabel 4. Pengujian pada Model dengan perbandingan dataset 80:20

NO	Gambar Uji Coba	Hasil prediksi			
		TP	TN	FP	FN
1		✓	-	-	-
2		✓	-	-	-
3		✓	-	-	-
4		✓	-	-	-
5		✓	-	-	-
6		✓	-	-	-
7		✓	-	-	-
8		-	✓	-	-
9		-	✓	-	-
10		-	-	✓	-
11		✓	-	-	-
12		-	✓	-	-
13		-	✓	-	-
14		-	✓	-	-
15		-	-	✓	-
	Total	**8**	**5**	**2**	**0**

Berdasarkan hasil pengujian pada Tabel 4, dapat dperoleh nilai dari *Precission, recall* dan *Accuracy* sebagai berikut :

Precision = (TP) / (TP+FP) = (8) / (8 + 2) = 0.8 = 80%

Recall = (TP) / (TP + FN) l = (8) / (8 + 0) = 1 = 100%

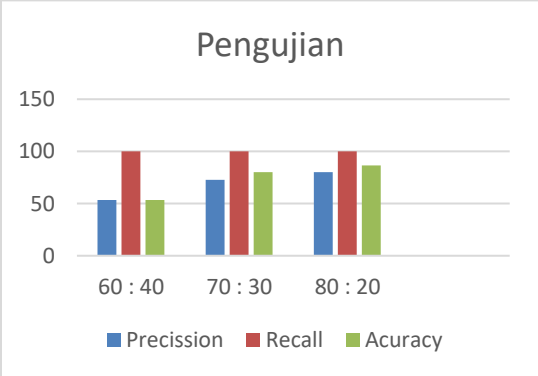
Accuracy = (TP + TN) / (TP+FP+FN+TN) = (8 + 5) / (8+2+0+5) = 0,866 = 86,6%

Pada Tabel 5 dapat dilihat hasil dari *Precision, Recall* dan *Accuracy* pada pengujian model yang dihasilkan berdasarkan hasil ujicoba menggunakan 3 skenario pembagian rasio dataset.

Tabel 5. Pengujian dengan *Confusion Matrix*

Rasio Dataset	*Precision*(%)	*Recall*(%)	*Accuracy*(%)
60 : 40	53,3	100	53,3
70 : 30	72,7	100	80
80 : 20	80	100	86,6

Berdasarkan hasil pengujian deteksi objek daun semanggi menggunakan input gambar melalui webcam secara *real – time* pada model yang dihasilkan dengan perbandingan rasio data *train* dan data *test* 60:40, 70:30 dan 80:20 didapatkan tingkat akurasi keberhasilan tertinggi ada pada perbandingan rasio 80:20 yaitu dengan akurasi 86.6 %, sedangkan yang terendah adalah 53,3% pada model data dengan rasio 60 :40.



Gambar 12. Grafik hasil pengujian

Pada Gambar 12 menampilkan grafik hasil pengujian bahwa pada perbandingan rasio data *train* dan data *test* 60:40 , 70:30, 80:20, memiliki tingkat akurasinya semakin bertambah tinggi, hal ini karena semakin banyak data *train* yang digunakan maka sistem akan lebih stabil dalam melakuan prediksi.

IV. Kesimpulan

Metode CNN-SSD yang digunakan dari penelitian ini berhasil diimplementasikan dengan baik untuk mendeteksi objek daun semanggi.Pengujian dilakukan pada model hasil *training* dengan menggunakan perbandingan rasio data *train* dan *test* 60:40, 70:30 dan 80:20. Hasil pengujian yang dilakukan pada 15 gambar menggunakan webcam secara real-time, didapatkan bahwa tingkat akurasi keberhasilan tertinggi ada pada perbandingan rasio data *train* dan data *test* 80:20 dengan rata-rata tingkat akurasi 86.6 %. Akurasi terendah pada rasio 60:40 sebesar 53,3%. Perbandingan tiap rasio data *train* dan data *test* memiliki tingkat akurasi yang semakin lama semakin besar. Hal ini membuktikan bahwa perbandingan rasio pada data *train* harus lebih banyak jika dibandingkan data *test*. Semakin banyak data yang ada pada data *train* maka sistem akan mendapatkan hasil prediksi yang lebih baik

Daftar Pustaka

[1] I. Amalia, Annisa. 2020. 10 Jenis Tanaman Obat-Obatan Yang Wajib Ada Di Rumah Anda. https://www.Sehatq.Com/Artikel/Jenis-Tanaman-Obat-Obatan-Yang-Wajib-Ada-Di-Rumah-Anda/ (Diakses Tanggal 2 Mei 2020).

[2] A. Arwan. 2018. 15 Manfaat Dan Khasiat Daun Semanggi Untuk Kesehatan. https://www.Atmago.Com/Posts/15-Manfaat-Dan-Khasiat-Daun-Semanggi-Untuk-Kesehatan_9383dc97-C447-4977-9f8a-6e8dc49aea67/ (Diakses Tanggal 9 Mei 2020).

[3] D. Rizky. 2019. Memahami Artificial Neural Network (ANN) Dengan R. https://Medium.Com/@16611129/Memahami-Artificial-Neural-Network-Ann-Dengan-R-5ecee7d1efbd/ (Diakses Tanggal 15 Mei 2020).

[4] L. Y. Bottou, L. Bengio & Haffner. 1998. "Gradient-Based Learning Applied To Document Recognition". *Proceedings Of The IEEE*. *86*(11), 2278-232.

[5] H. Lars. 2018. A Beginner's Guide To Object Detection. https://www.Datacamp.Com/Community/Tutorials/Object-Detection-Guide (Diakses Tanggal 15 April 2020).

[6] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andretto, H. Adam. 2017. "Mobilenets: Efficient Convolutional Neural Network For Mobile Vision Applications".

[7] I.W. Suartika,.A.Y. Wijaya, R. Soelaiman. 2016. "Klasifikasi Citra Menggunakan Convolutiona Neural Network". *Jurnal Teknik Its*. *5*(1), 2337-3539.

[8] Soeleman. 2017. "Instalasi Python Lebih Mudah Dengan Anaconda". https://www.Codepolitan.Com/Instal-Python-Dengan-Anaconda-58a79fee367c0/. (Diakses Tanggal 15 April 2020).

[9] Doavers, 2018. Apa Itu Ekstraksi Fitur Pada Citra Digital. https://www.Doavers.Com/Blog/Apa-Itu-Ekstraksi-Fitur-Pada-Citra-Digital/ (Diakses Tanggal 20 Juni 2020).

[10] A. Arfina, E. Utami, E. Pranomo. 2017. "Modifikasi Default-Boxes Pada Model Ssd Untuk Meningkatkan Keakuratan Deteksi". *Jurnal It Cida, 3*[2], 2477-8125.

[11] R. D. Syarifah. 2018. "Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network". *Skripsi. Fakultas Matematika Dan Ilmu Pengetahuan Alam. Statistika. Universitas Islam Indonesia. Yogyakarta.*

[12] A. Lazaro. 2017. "Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV". *Diss. Institut Teknologi Sepuluh Nopember*.

[13] Tensorflow. 2017. "Tensorflow Object Detection Api". Https://Github.Com/Tensorflow/Models /. (Diakses Tanggal 15 April 2020).

[14] W. L. D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A. C. Berg. 2015. *Ssd:Single Shot Multibox Detector*.

[15] U. Ema, S. Raharjo. 2004. "Logika, Algoritma Dan Impelementasinya Dalam Bahasa Python Di Gnu/Linux". Yogyakarta: Andi Offset.

[16] K.S. Nugroho. 2019. Confusion Matrix Untuk Evaluasi Model Pada Supervised Learning. https://Medium.Com/@Ksnugroho/Confusion-Matrix-Untuk-Evaluasi-Model-Pada-Unsupervised-Machine-Learningbc4b1ae9ae3f.