DETEKSI TANAMAN TEBU PADA LAHAN PERTANIAN MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK*

Muhammad Alfin Jimly Asshiddiqie[1)], Basuki Rahmat[2)], Fetty Tri Anggraeny[3)]
E-mail : [1)] alfinjimly@gmail.com, [2)] basukirahmat.if@upnjatim.ac.id ,
[3)] fettyanggraeny.if@upnjatim.ac.id

[1,2,3] Teknik Informatika, Fakultas Ilmu Komputer, UPN Veteran Jawa Timur

Abstrak

Indonesia merupakan negara agraris yang memiliki lahan pertanian yang luas. Kondisi tersebut sesuai untuk produksi tanaman padi dan palawija seperti jagung, singkong, kedelai, kacang, tebu, cabai, dan lain sebagainya. Sebagai upaya untuk lebih memperkenalkan tanaman pertanian kepada masyarakat, maka diperlukan sebuah model yang dapat mengenali bentuk dan jenis tanaman pertanian di Indonesia. Hal ini dapat diwujudkan dengan memanfaatkan Drone, yaitu teknologi pesawat tanpa awak yang dapat digunakan untuk mengindentifikasi jenis tanaman melalui udara. Penelitian ini dilakukan untuk mengembangkan sistem deteksi jenis tanaman berbasis drone dengan menerapkan algoritma CNN (Convolutional Neural Network) dengan menggunakan YOLO (You Only Look Once). Hasil penelitian menunjukkan bahwa metode CNN berhasil untuk mendeteksi tebu dengan cukup baik dengan menghasilkan rata-rata nilai confidence sebesar 95% pada pengujian video. Pengujian menggunakan pada nilai threshold 0.1, menghasilkan skor precision sebesar 1.00, skor recall sebesar 0.95 dan skor accuracy sebesar 0.95 pada tebu.

Kata kunci: Lahan Pertanian*, Drone,* Deteksi Tanaman*, Convolutional Neural Network (CNN),* Algoritma *YOLO (You Only Look Once)*

1. PENDAHULUAN

Indonesia merupakan negara agraris di mana sebagian besar penduduknya hidup dari hasil bercocok tanam atau bertani, sehingga pertanian merupakan sektor yang memegang peranan penting dalam kesejahteraan kehidupan penduduk Indonesia. Indonesia memiliki lahan pertanian yang luas, sebagian besar dengan kondisi iklim kering, dan kondisi tersebut sesuai untuk produksi tanaman padi dan palawija seperti jagung, singkong, kedelai, kacang, tebu, cabai, dan lain sebagainya. Sebagai upaya untuk lebih memperkenalkan tanaman pertanian kepada masyarakat, maka diperlukan sebuah model yang dapat mengenali bentuk dan jenis tanaman pertanian di Indonesia. Dengan memanfaatkan ilmu komputasi yang memungkinkan komputer untuk mengambil informasi dari suatu citra digital dalam pengenalan objek secara otomatis, maka diharapkan model tersebut dapat menjadi salah satu solusi dalam mengenali bentuk tanaman pertanian dan dapat memudahkan lembaga yang terkait untuk melakukan klasifikasi terhadap tanaman pertanian.

Salah satu solusi yang dapat dilakukan adalah klasifikasi jenis tumbuhan melalui udara. Hal ini tentu dapat dilakukan dengan memanfaatkan teknologi drone atau pesawat tanpa awak. Pesawat ini dikendalikan secara otomatis melalui program komputer [1]. Dalam hal ini, drone dapat dimanfaatkan dengan membawa kamera yang digunakan untuk mengambil citra foto maupun video untuk memantau keadaan lingkungan sekitar. Selanjutnya, citra maupun video tersebut dapat dianalisa untuk mengidentifikasi jenis tanaman tersebut. Salah satu cara mendeteksi jenis tumbuhan citra digital adalah dengan melakukan teknik peningkatan kualitas citra (image enhancement). Selanjutnya, citra

hasil dari image enhancement dapat dilatih dengan memanfaatkan algoritma deep learning yaitu algoritma YOLO (You Only Look Once).

Algoritma YOLO (You Only Look Once) adalah algoritma deep learning yang memanfaatkan jaringan syaraf konvolusional (CNN) dalam mendeteksi objek. Algoritma ini akan membagi citra ke dalam grid berukuran s x s yang kemudian pada tiap grid akan memprediksi bounding box serta peta kelas masing-masing grid. Apabila pada satu grid terprediksi objek, maka pada grid tersebut akan diprediksi bounding box yang mengelilingi objek tersebut. Nilai confidence akan dihitung pada masing-masing bounding box yang kemudian akan diseleksi berdasarkan nilai yang didapat [2].

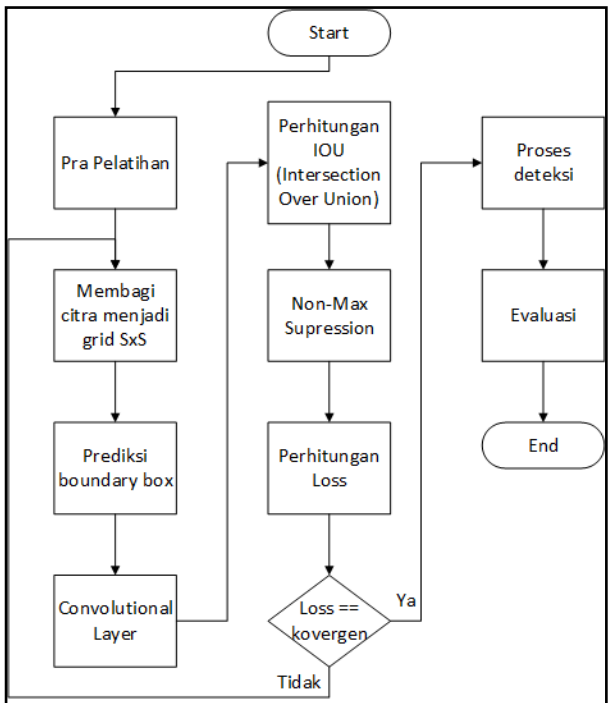
2. METODOLOGI

2.1 Analisa Data

Data yang digunakan dalam penelitian ini adalah 350 citra data tebu dan video hasil perekaman tanaman lahan pertanian yang diambil secara mandiri dengan drone. Video yang digunakan sebagai data uji berjumlah 3 video dengan masing-masing berdurasi 10-30 detik dengan framerate 20 fps.

2.2 Analisa Sistem

Analisa sistem penelitian ini dijelaskan pada Gambar 1 berikut :



Gambar 1. Tahap penelitian

Pada gambar 1 menjelaskan alur jalannya sistem pada penelitian ini. Proses yang dilakukan melalui 3 tahapan diantaranya pra pelatihan, pelatihan, dan deteksi atau pengujian, kemudian membagi citra menjadi grid S x S dan prediksi boundary box. Setelah itu, dilakukan proses perhitungan Intersection Over Union (IOU), non-max supression, dan perhitungan loss. Apabila loss yang dihasilkan belum konvergen, kembali ke proses CNN, jika sudah konvergen, akan dilanjutkan pada tahap deteksi atau pengujian. Proses deteksi akan menghasilkan output bounding box pada video yang telah diuji dan terakhir dilakukan proses evaluasi.

2.3 Algoritma YOLO *(You Only Look Once)*

Algoritma YOLO merupakan algoritma deep learning untuk deteksi objek yang menggunakan pendekatan berbeda dari algoritma lain, yaitu menerapkan sebuah jaringan syaraf tunggal pada keseluruhan citra. YOLO mendeteksi sebuah objek dalam beberapa tahap [2] :

1. Membagi citra dalam region/grid berukuran sxs. Grid-grid tersebut bertanggung jawab untuk mendeteksi objek. Pada gambar 2, tiap grid juga akan diprediksi bounding box beserta nilai confidence. Nilai confidence ini menunjukkan seberapa yakin bounding box tersebut berisi objek dan seberapa akurat prediksinya. Nilai confidence diperoleh melalui persamaan :

$$Conf(class) = \Pr(class) \times IOU_{Pred}^{Truth} \quad (1)$$

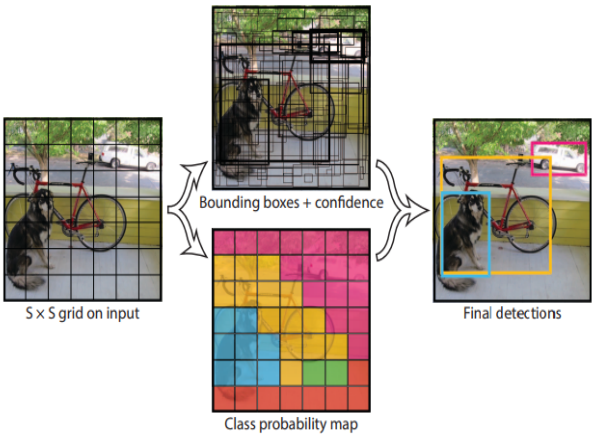
$Pr(class)$ adalah probabilitas objek yang muncul dalam suatu region dan IOU_{Pred}^{Truth} adalah rasio tumpang tindih (Intersection Over Union) antara kotak prediksi dan kotak ground truth. Pred adalah luas area dalam kotak prediksi, Truth adalah area dalam ground truth. Makin besar nilai IOU, maka makin tinggi tingkat akurasi pendeteksiannya [3].

2. Tiap bounding box memiliki 5 nilai informasi yaitu x,y,w,h dan c. Nilai x dan y adalah koordinat titik tengah bounding box yang terprediksi, nilai w dan h adalah rasio ukuran lebar dan tinggi relatif terhadap grid, dan c adalah nilai confidence bounding box tersebut.

3. Pada algoritma YOLO, tiap grid akan memprediksi nilai class probabilitas jika diprediksi terdapat objek di dalamnya. Saat pengujian, YOLO akan mengkalikan nilai class probability dengan nilai confidence dari bounding box.

$$Pr\ (Class_i|Object) \times \Pr(Object) \times IOU_{Pred}^{Truth} = Pr\ (Class_i) \times IOU_{Pred}^{Truth} \quad (2)$$

Sehingga menghasilkan nilai confidence kelas secara spesifik pada tiap bounding box. Nilai ini menunjukkan class probability yang muncul pada bounding box dan seberapa akurat bounding box memprediksi sesuai dengan objek [2].



Gambar 2. YOLO [2]

2.4 Arsitektur CNN pada YOLO

Arsitektur jaringan syaraf konvolusional yang digunakan pada penelitian ini adalah arsitektur YOLOv2 atau YOLO9000 [4]. Berikut adalah arsitektur yang digunakan :

Tabel 1. Tabel Arsitektur CNN pada YOLOv2

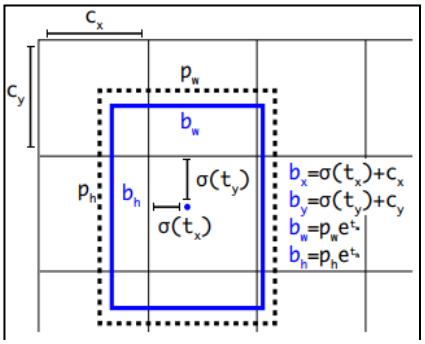
Input layer	***Ukuran dimensi***	***Padding***	***Output***	***Filter***	***Activation***
Input	3x3	1	416x416	32	Leaky ReLU
Conv 1 (C1)	2x2	0	208x208	32	
Max Pooling 1(MP1)	3x3	1	208x208	64	Leaky ReLU
C2	2x2	0	104x104	64	
MP2	3x3	1	104x104	128	Leaky ReLU
C3	1x1	0	104x104	64	Leaky ReLU
C4	3x3	1	104x104	128	Leaky ReLU
C5	2x2	0	52x52	128	
MP3	3x3	1	52x52	256	Leaky ReLU
C6	1x1	0	52x52	128	Leaky ReLU
C7	3x3	1	52x52	256	Leaky ReLU
C8	2x2	0	26x26	256	
MP4	3x3	1	26x26	512	Leaky ReLU
C9	1x1	0	26x26	256	Leaky ReLU
C10	3x3	1	26x26	512	Leaky ReLU
C11	1x1	0	26x26	256	Leaky ReLU
C12	3x3	1	26x26	512	Leaky ReLU
C13	2x2	0	13x13	512	
MP5	3x3	1	13x13	1024	Leaky ReLU
C14	1x1	0	13x13	512	Leaky ReLU
C15	3x3	1	13x13	1024	Leaky ReLU
C16	1x1	0	13x13	512	Leaky ReLU
C17	3x3	1	13x13	1024	Leaky ReLU
C18	3x3	1	13x13	1024	Leaky ReLU
C19	3x3	1	13x13	1024	Leaky ReLU
C20			26x26	512	
Concat	1x1	0			Leaky ReLU
C21	2x2		13x13	256	
Flatten			13x13	1280	
Concat	3x3	1	13x13	1024	Leaky ReLU
C22	1x1	0	13x13	40	Linear
C23	3x3	1	416x416	32	Leaky ReLU

Pada Tabel 1, terlihat bahwa arsitektur CNN yang digunakan terdiri dari 23 konvolusi layer, 5 pooling layer dan menghasilkan feature map berukuran 13x13x40. Fungsi aktivasi menggunakan Leaky ReLU untuk tiap convolusi kecuali konvolusi terakhir, karena akan digunakan sebagai penentu klasifikasi. Batch normalization ditambahkan pada convolution layer untuk meningkatkan akurasi pelatihan sehingga mempercepat nilai loss menuju konvergen.

2.5 Prediksi Bounding Box

Untuk memprediksi bounding box, YOLOv2 akan memprediksi koordinat titik tengah bounding box dengan relatif terhadap lokasi grid, dengan menggunakan k-means clustering dengan menggunakan nilai k = 5 yang memberikan perbandingan nilai recall dan kompleksitas model yang baik.. Sedangkan untuk menghitung jarak euclidean YOLOv2 menggunakan pendekatan IOU dengan rumus [4] :

$$d(box, centroid) = 1 - IOU(box, centroid) \quad (3)$$



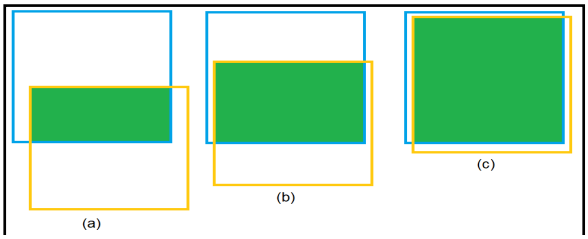
Gambar 3. Prediksi Bounding Box (Redmon & Farhadi, 2016)

Pada Gambar 3, menunjukkan prediksi bounding box pada YOLOv2. YOLOv2 memprediksi bounding box dari nilai-nilai anchor box yang terbentuk. Nilai bx diperoleh dari nilai sigma tx(ordinat x prediksi) dan nilai panjang dari cx (cell x). Nilai by diperoleh dari nilai sigma ty(ordinat y prediksi) dan nilai panjang dari cy (cell y). Nilai bw dan bh, diperoleh dari nilai lebar dan nilai tinggi bounding box. Untuk prediksi objek diperoleh dari nilai sigma to.

2.6 *Intersection Over Union (IOU)*

IOU menghitung luas area yang berpotongan lalu membaginya dengan luas area gabungan antar 2 bounding box. Nilai IOU digunakan untuk menentukan kesesuaian bounding box yang diprediksi dengan luas objek yang sesungguhnya pada citra (ground truth). Gambar 4 merupakan contoh dari IOU. IOU dapat dihitung dengan persamaan :

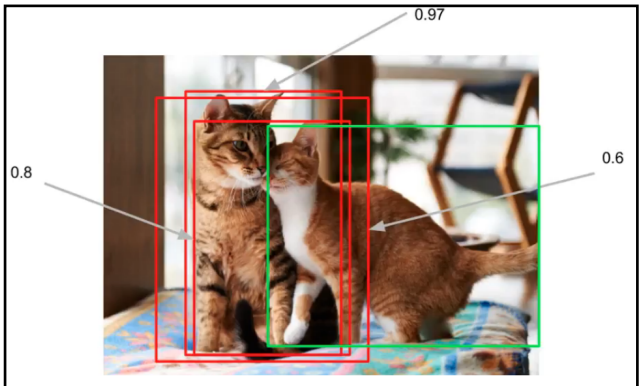
$$IOU = \frac{Area\ Overlap}{Area\ Union} \quad (7)$$



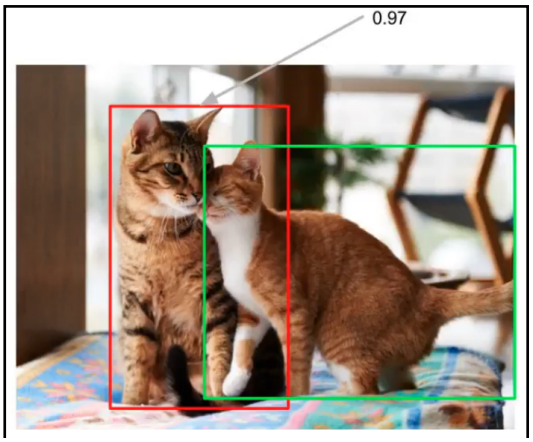
Gambar 4. Nilai IoU (a). buruk, (b). baik, (c). sangat baik

2.7 *Non-Max Supression*

Non-Max Suppression atau NMS berhubungan dengan algoritma objek deteksi untuk memfilter bounding box. NMS membuat seleksi berdasarkan IoU dengan mengurangi bounding box yang muncul secara berlebihan. Seleksi dilakukan hingga tersisa satu saja bounding box dan memiliki nilai konfiden tertinggi.



Gambar 5. Non-max Suppression (1)



Gambar 6. Non-max Suppression (2)

Pada gambar 5, terdapat 3 bounding box berwarna merah dan masing masing memiliki nilai IoU 0.8, 0.97, dan 0.6. NMS menyeleksi nilai tertinggi dan menghapus bounding box yang memiliki nilai IoU yang lebih kecil. Sehingga pada gambar 6, bounding box yang akan muncul adalah box yang memiliki nilai tertinggi yaitu 0.97 dan menghapus 2 bounding box lainnya. Box berwarna hijau tidak termasuk karena tidak memiliki nilai IoU yang tinggi terhadap box warna merah.

2.1 *Loss Function*

Untuk mengevaluasi *bounding box* yang terprediksi pada citra dapat menggunakan rumus perhitungan *Loss Function* sebagai berikut (Redmon, Divvala, Girshick, & Farhadi, 2016) :

$$\begin{aligned} Loss\ function = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c) \right)^2 \end{aligned} \quad (8)$$

Pada baris pertama persamaan 8 merupakan perhitungan evaluasi dari prediksi koordinat titik tengah pada (x,y) *bounding box* jika terdapat objek pada sebuah grid, dengan (x_i , y_i) adalah koordinat objek yang sebenarnya (*ground truth*) dan ($\hat{x_i}$, $\hat{y_i}$) adalah koordinat *bounding box* yang terprediksi. Pada baris kedua, merupakan perhitungan evaluasi dari prediksi lebar dan tinggi (*widht* dan *height*) *bounding box*, dengan (w_i , h_i) adalah luas objek yang sebenarnya(*ground truth*) dan ($\hat{w_i}$, $\hat{h_i}$) adalah nilai *bounding box* yang terprediksi. Untuk baris ketiga merupakan perhitungan evaluasi nilai *confidence* dari *bounding box* yang terprediksi. Sedangkan pada baris keempat, merupakan evaluasi untuk prediksi kelas-kelas objek yang terdeteksi.

3. HASIL DAN PEMBAHASAN

Proses training menggunakan konfigurasi yolov2-voc.cfg dan bobot yolov2-voc.weights. Tahap pelatihan dilakukan hingga loss mencapai konvergen dengan melakukan epoch sebanyak 200. Maka, didapatkan hasil sebagai berikut :

3.1 Evaluasi Hasil Data Video

Gambar 7. Hasil Deteksi pada Video

Pada pengujian yang dilakukan dengan 3 data video pada nilai threshold 0.1, didapati bounding box dapat terbentuk dari sekeliling objek tanaman dengan menampilkan nilai confidence-nya pada tiap frame seperti pada gambar 7. Kemudian membagi video menjadi frame per frame untuk melakukan perhitungan confusion matrix. Total frame yang dihasilkan oleh 1 video yaitu 800 frame atau 800 citra.

3.2 Evaluasi Confusion Matriks

Pada tabel 2 dijelaskan perhitungan confusion matrix tebu pada 800 citra dan menggunakan parameter threshold 0.1.

Tabel 2. Confusion matrix tebu 1

	Predicted		
	Class	Tebu	Non-tebu
Actual	Tebu	423	377
	Non-tebu	0	0

Tabel 3. Confusion matrix tebu 2

	Predicted		
	Class	Tebu	Non-tebu
Actual	Tebu	760	40
	Non-tebu	0	0

Tabel 4. Confusion matrix tebu 3

	Predicted		
	Class	Tebu	Non-tebu
Actual	Tebu	160	630
	Non-tebu	0	0

Dari tabel 3 maka dilakukan perhitungan *precision, recall, accuracy* :

Perhitungan *precision* : $\frac{760}{760+0} = \frac{760}{760} = 1.0$

Perhitungan *recall* : $\frac{760}{760+40} = \frac{760}{800} = 0.95$

Perhitungan *accuracy* : $\frac{760+0}{760+0+0+40} = \frac{760}{800} = 0.95$

Tabel 5. Confusion matrix tebu 2

	Data	*Precision*	*Recall*	*Accuracy*
Tebu	1	1.00	0.53	0.53
	2	1.00	0.95	0.95
	3	1.00	0.21	0.21

Pada tabel 5 data ke-2, deteksi tebu dengan hasil prediksi tebu (TP) sebanyak 760 citra dan deteksi tebu dengan hasil prediksi non-tebu (FN) sebanyak 40 citra. Total yaitu 800 citra. Dari data tabel 2 menghasilkan nilai precision 1.0, nilai recall 0.95 dan accuracy sebesar 0.95. Nilai precision mengukur seberapa akurat prediksi sebuah model. Nilai recall mengukur seberapa bagus model menemukan nilai positif.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat diambil beberapa kesimpulan di antaranya :

1. Algoritma CNN dapat mendeteksi objek tebu pada video ke 2 dengan cukup baik dengan mendapat nilai *confidence* dengan nilai terendah 20.03% dan nilai tertinggi 85.70%
2. Dengan menggunakan nilai *threshold* 0.10, didapatkan hasil deteksi tebu dengan skor *precision* sebesar 1.0, *recall* sebesar 0.95 , *accuracy* sebesar 0.95. serta skor mAP sebesar 56.37%.

5. DAFTAR RUJUKAN

[1] Utomo, B. (2017). Drone Untuk Percepatan Pemetaan Bidang Tanah. Media Komunikasi Geografi, Vol 18, No. 2 Jurusan Pendidikan Geografi, Universitas PGRI Palembang, 146-155.

[2] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition, 779-788.

[3] Lan, W., Dang, J., Wang, Y., & Wang, S. (2018). Pedestrian Detection Based on YOLO Network Model. Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, 1547-1551.

[4] Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. arXiv:1612.08242v1, 1-9.