

Implementasi Algoritma You Only Look Once (YOLO) Untuk Object Detection Sarang Orang Utan

I Made Dwijaya Maleh¹, Rony Teguh², Abertun Sagit Sahay³, Simon Okta⁴, Muhammad Porkab Pratama⁵

^{1,2,3,4,5} Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangkaraya
Jl. Yos Sudarso, Palangka Raya, Kalimantan Tengah, Indonesia

e-mail: ¹madej250@gmail.com, ²ronnyteguh@gmail.com, ³abertun@it.upr.ac.id,
⁴simonokta@gmail.com, ⁵muhammadporkabpratama@gmail.com

Informasi Artikel

Diterima: 12-09-2022

Direvisi: 15-01-2023

Disetujui: 19-01-2023

Abstrak

Hutan gambut di Kalimantan adalah salah satu ekosistem hidrologi penting di dunia yang memainkan peran vital dalam menjaga keseimbangan lingkungan. Hutan ini memfasilitasi siklus air dan membantu mengurangi perubahan iklim dengan menyerap karbon. Beragam spesies tanaman dan hewan unik hidup di hutan gambut dan sangat penting bagi kesehatan lingkungan. Aliran air dari hutan gambut menjadi sumber air bagi pemukiman dan pertanian. Hutan gambut juga berfungsi sebagai penyimpan karbon besar dan dapat menyimpan jumlah karbon setara dengan emisi global selama beberapa tahun. Untuk melindungi hutan gambut sebagai ekosistem hidrologi dan penyerap karbon. Orangutan adalah spesies penting yang harus dilindungi untuk menjamin keanekaragaman hayati di dunia. Keberadaan orangutan adalah indikator lingkungan yang sehat dan membutuhkan habitat alami yang luas dan kaya. Sarang-sarang yang ada di hutan menunjukkan lingkungan masih alami dan bisa memenuhi kebutuhan orangutan. Oleh karena itu, menjaga dan memelihara sarang orangutan sangat penting untuk melestarikan spesies dan ekosistem hutan. Implementasi algoritma YOLO (You Only Look Once) merupakan salah satu upaya untuk mendeteksi sarang orangutan di hutan Kalimantan. Algoritma ini memprediksi kotak pembatas dan probabilitas kelas pada gambar penuh dalam satu tangkapan. Dengan 1970 citra latih dan 414 citra berlabel sarang orangutan, hasil training menunjukkan Precision 0,973, Recall 0.949, mAP_{0.5} 0.969, dan mAP_{0.5:0.95} 0.630. Model ini membutuhkan 58 jam untuk menyelesaikan 217 epoch dan menghasilkan akurasi deteksi objek yang tinggi model menunjukkan hasil deteksi jumlah sarang orangutan dengan akurasi rata-rata 99.9%.

Kata Kunci: You Only Look Once (YOLO), Hutan gambut Kalimantan, Orang Utan

Abstract

The peat forest in Kalimantan is one of the world's important hydrological ecosystems that play a vital role in maintaining environmental balance. This forest facilitates the water cycle and helps reduce climate change by absorbing carbon. A variety of unique plant and animal species live in the peat forest and are crucial for environmental health. The water flow from the peat forest becomes a source of water for settlements and agriculture. The peat forest also functions as a large carbon store and can store an amount of carbon equivalent to global emissions for several years. To protect the peat forest as a hydrological ecosystem and carbon absorber. Orangutans are important species that must be protected To ensure biodiversity in the world. The presence of orangutans is a sign of a healthy environment and requires a large and rich natural habitat. Nests found in the forest indicate that the environment is still natural and can meet the needs of orangutans. Therefore, preserving and maintaining orangutan nests is very important for conserving species and forest ecosystems. The implementation of YOLO (You Only Look Once) algorithm is one effort to detect orangutan nests in the Kalimantan forest. This algorithm predicts the bounding box and class probability on a full image in one shot. With 1970 training images and 414 labeled orangutan nest images, the training results showed Precision 0.973, Recall 0.949, mAP_{0.5} 0.969, and mAP_{0.5:0.95} 0.630. The model took 58 hours to complete 217 epochs and produced high object detection accuracy. The model shows results of detecting the number of orangutan nests with an average accuracy of 99.9%.

Keywords: You Only Look Once (YOLO), Kalimantan peat forest, Orang Utan



1. Pendahuluan

Orangutan merupakan satu-satunya kera besar yang hidup di Asia, sementara tiga kerabatnya, yaitu; gorila, simpanse, dan bonobo hidup di Afrika. Kurang dari 20.000 tahun yang lalu orangutan dapat dijumpai di seluruh Asia Tenggara, dari Pulau Jawa di ujung selatan sampai ujung utara Pegunungan Himalaya dan Cina bagian selatan. Akan tetapi, saat ini jenis kera besar itu hanya ditemukan di Sumatera dan Borneo (Kalimantan), 90% berada di Indonesia dan saat ini populasi orangutan di kedua pulau tersebut telah dibedakan menjadi dua spesies yaitu orangutan Sumatera (*Pongo abelii*) dan orangutan Kalimantan (*Pongo pygmaeus*) (Rowe, 1996 ; Groves, 1999 ; Supriatna dan Wahyono, 2000). Di alam, orangutan dijumpai hidup pada habitat hutan dataran rendah secara semi soliter dan arboreal. Sebagai satwa *frugivora*, orangutan ditemukan banyak mengkonsumsi buah sebagai makanan utamanya (Rowe, 1996 ; Whitten dan Compost, 1998 ; Meijaard dkk., 2001). Orangutan dapat dijadikan 'umbrella species' (spesies payung) untuk meningkatkan kesadaran konservasi masyarakat. Kelestarian orangutan menjamin kelestarian hutan yang menjadi habitatnya, sehingga diharapkan kelestarian makhluk hidup lain ikut terjaga pula. Sebagai pemakan buah, orang utan merupakan agen penyebar biji yang efektif untuk menjamin regenerasi hutan. Orangutan juga sangat menarik dari sisi ilmu pengetahuan karena kemiripan karakter biologi satwa itu dengan manusia. Sebagai satu-satunya kera besar yang hidup di Asia, orangutan memiliki potensi menjadi ikon pariwisata untuk Indonesia.

Artificial Intelligence atau Kecerdasan Buatan adalah ilmu yang mempelajari kemampuan dari sebuah sistem untuk mengartikan data eksternal dengan benar, untuk dipelajari dan menggunakan pembelajarannya sehingga tercapai sebuah tujuan maupun tugas melalui adaptasi yang *fleksibel* (Kaplan & Haenlein, 2019). Salah satunya adalah YOLO, YOLO (*You Only Look Once*) adalah pendekatan terbaru dalam dunia pendeteksi objek. Yang mana untuk memenuhi persyaratan deteksi dengan akurasi maupun kecepatan tinggi dan operasi secara langsung (*real-time*).

$$\phi(x) = \{x, \text{jika } x > 0 \text{ or } 0.1x, \text{yang lain}$$

$\phi(x)$ adalah sebuah fungsi yang digunakan untuk menentukan nilai dari x . Fungsi ini memiliki dua kondisi jika $x > 0$, maka $\phi(x) = x$ atau jika $x \leq 0$, maka $\phi(x) = 0.1x$ Kondisi pertama menyatakan bahwa jika nilai x lebih besar dari 0, maka nilai $\phi(x)$ sama dengan x itu sendiri. Kondisi kedua menyatakan bahwa jika

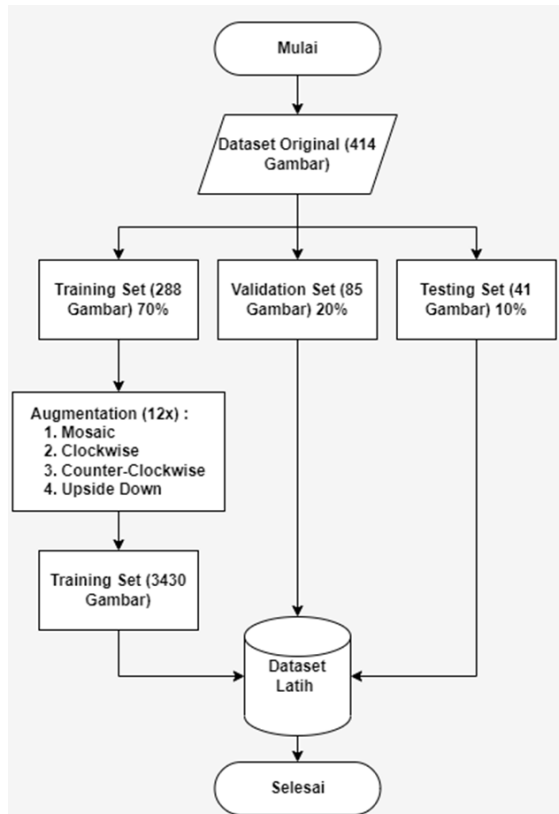
nilai x tidak lebih besar dari 0, maka nilai $\phi(x)$ sama dengan 0.1 kali x atau dengan nilai lainnya.

YOLO merupakan metode detektor dengan model terpadu (*unified*), yang mana dengan jaringan saraf tunggal (*single neural network*) dapat memprediksi kotak pembatas (*Bounding Box*) dan probabilitas kelas secara langsung dalam satu gambar penuh pada sekali tangkapan. Pada model YOLO, dapat memproses gambar inputan hingga pada 45 FPS (*frame per second*), bahkan dengan versi jaringan neural yang lebih kecil lagi, yakni Fast YOLO dapat memproses hingga 155 FPS dan menjadi algoritma tercepat dalam perbandingan dengan algoritma pendeteksian real-time lainnya berada jauh di atas metode-metode non real-time seperti *Fast R-CNN* dan *Faster R-CNN* yang hanya berkisar 0.5 FPS dan 7 FPS. Hal ini dikarenakan YOLO menerapkan teknik perhitungan single shot detection, dimana CNN hanya dijalankan sekali saja dalam proses deteksi objek. Ini berbeda dari metode lainnya, seperti R-CNN dan *its variants*, yang menjalankan CNN beberapa kali untuk setiap region proposal.

Dalam penelitian ini, maka penelitian ini akan berfokus pada penerapan algoritma YOLO (*You Only Look Once*) dalam pendeteksian sarang orang utan dengan judul yang diharapkan mampu membantu dalam konservasi orang utan serta menjaga keanekaragaman hayati yang ada di Indonesia.

2. Metode Penelitian

Pada penelitian ini citra gambar data test yang terdapat di dataset dimana model yang sudah di latih tidak pernah melihatnya yang nantinya akan digunakan sebagai inputan yang akan diproses apakah terdapat Sarang Orang Utan pada citra gambar tersebut. Artificial intelegent adalah proses selanjutnya. Kecerdasan buatan (*artificial intelligence*) merupakan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan hal yang dalam pandangan manusia adalah cerdas. (Detty Purnamasari, 2013) lalu dapatkan *bounding box* dan prediksi kelas dalam output. Selanjutnya, *dataset* akan di-*training*. Tahapan ini, menggunakan metode YOLO. YOLO merupakan jaringan untuk mendeteksi objek. Tugas deteksi objek terdiri dalam menentukan lokasi pada gambar dimana objek tertentu hadir serta bahwa objek tersebut apakah merupakan Sarang Orang Utan.



Sumber : Penulis (2022)

Gambar 1. Pembagian Dataset

Pengumpulan Data yang digunakan dalam penelitian ini antara lain gambar hutan yang terdapat sarang orang utan. Pengambilan data dilakukan saat kondisi langit cerah dan posisi lurus menghadap permukaan hutan. Pembuatan model dengan metode YOLO membutuhkan *dataset* untuk dilakukan pelatihan. *Dataset* yang digunakan dalam penelitian ini adalah dengan meng-input citra gambar sebanyak 1970 citra citra hutan dari atas terdapat label Sarang Orang Utan yang di ambil oleh *drone* gambar hanya terdapat 414 citra gambar yang terdapat sarang Orang Utan Data ini terbagi menjadi 70% atau 288 gambar citra data *train*, 20 % atau 85 gambar citra data *validation* dan 10% 41 gambar citra data *test*.

Pada tahap *preprocessing*, data yang diperoleh dari pengumpulan data masih berupa data mentah yang perlu diolah untuk memperjelas objek dari suatu citra. *Preprocessing* data meliputi pelabelan dan perubahan ukuran citra. Pelabelan dilakukan untuk memberikan label pada objek Sarang Orang Utan dan menyimpan informasi citra. Pelabelan dilakukan dengan menggunakan *LabelImg* yang akan menghasilkan file dengan format *.txt* yang berisi informasi seperti path citra, ukuran, nama label, dan nilai batas-batas *Bounding Box*. Selain itu, dilakukan juga teknik *Data Augmentation* untuk mutipikasi data atau

menambah jumlah data dengan menambahkan salinan yang sedikit dimodifikasi dari data yang sudah ada atau data sintesis yang baru dibuat dari data yang ada. *Data Augmentation* digunakan untuk meningkatkan jumlah data dan melatih model dengan data yang lebih banyak dan bervariasi.

Pelatihan data (*training*) akan dilakukan setelah *preprocessing* data untuk mempelajari suatu pola agar dapat menghasilkan pengenalan objek Sarang Orang Utan pada saat mendeteksi objek di sebuah citra. Citra objek Orang Utan yang akan digunakan sebagai data pelatihan/training. Pelatihan akan dilakukan dengan model arsitektur YOLO (*You Only Look Once*) dengan file bobot dan konfigurasi yang didapat dari *framework darknet*. Model ini memiliki jaringan saraf yang terdiri dari lapisan konvolusional. Secara bergantian lapisan konvolusi kernel mengurangi fitur dari lapisan sebelumnya (*max pooling*) diikuti oleh lapisan konvolusi. Proses akan diulangi apabila gagal, sebaliknya apabila berhasil maka dilanjutkan ke metode pendukung berikutnya.

Perancangan dan Implementasi Deteksi Sarang Orang Utan Pada penelitian ini, metode yang digunakan untuk mendeteksi adalah metode YOLOv5 dengan menggunakan pre-trained weight dan model dari darknet open source YOLO. Sehingga perlu untuk menyesuaikan kebutuhan hardware dan software dalam mengimplementasikan YOLOv5 dari darknet untuk dataset yang digunakan dalam penelitian ini.

$$IoU = \frac{\text{Area of overlap}}{\text{Area of union}} \quad (1)$$

IoU (Intersection over Union) merupakan rasio antara luas wilayah yang tumpang tindih (*overlap*) dari dua kotak pembatas (*bounding box*) dengan luas wilayah gabungan (*union*) dari kedua kotak pembatas tersebut.

$$AP = \sum_{k=0}^{k=n-1} [\text{Recalls}(k) - \text{Recalls}(k+1)] * \text{Precisions}(k) \quad (2)$$

AP (Average Precision) adalah metrik yang digunakan untuk mengevaluasi hasil deteksi objek dalam suatu sistem deteksi. $\sum_{(k=0)^{(k=n-1)}$ adalah notasi *sigma*, yang berarti penjumlahan dari setiap nilai *k* yang dimulai dari 0 hingga *n-1*. *Recalls(k) - Recalls(k+1)* * *Precisions(k)* adalah perbedaan antara *recall* pada *k* dengan *recall* pada *k+1* dikali dengan presisi pada *k*.

$$mAP = \frac{1}{n} \sum_{k=0}^{k=n} APk \quad (3)$$

mAP (*Mean Average Precision*) adalah metrik yang digunakan untuk mengevaluasi hasil deteksi objek dalam suatu sistem deteksi. $1/n$ adalah rasio dari jumlah kelas objek yang diuji dengan jumlah total kelas objek dan $\sum (k=1)^n AP_k$ adalah penjumlahan dari setiap nilai AP yang dihasilkan dari setiap kelas objek dari $k = 1$ sampai $k = n$.

Validasi kinerja program dilakukan dengan menghitung nilai IoU antara prediksi kotak pembatas dengan kotak pembatas yang sebenarnya dan menghitung akurasi klasifikasi objek dari *Recall* dan *Precision* kemudian menghasilkan *Average Precision (AP)* dan *Mean Average Precision (mAP)*.

3. Hasil dan Pembahasan

Hasil Pembahasan Dan pengujian dan dilakukan oleh panulis akan di jelaskan pada bagian ini. Tujuan Dari bab ini adalah untuk mengetahui tingkat keberhasilan terhadap perancangan sistem yang telah di kerjakan. Pengujian yang di lakukan meliputi, pengujian tingkat akurasi deteksi sarang orang hutan. Terdapat dua istilah penting dalam pembangunan model machine learning yaitu: training dan testing. Training adalah proses konstruksi model dan testing adalah proses menguji kinerja model pembelajaran.

Anotasi Objek

Anotasi atau pelabelan objek adalah proses pembuatan label pada gambar dengan cara memberikan kotak pembatas (*Bounding Box*) beserta nama kelas pada objek. Proses anotasi citra menggunakan *software Labellmg*. Pelabelan ini menggunakan anotasi dalam format anotasi YOLO.

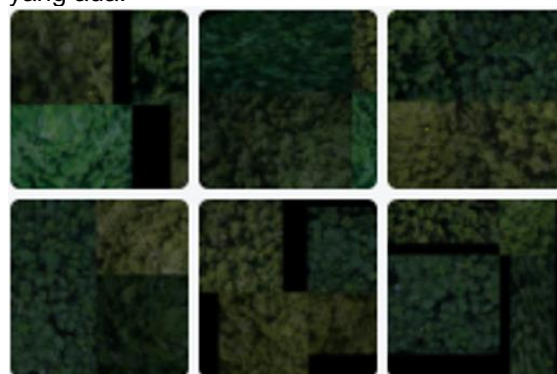
$$y = \begin{bmatrix} P(Object)_1 \\ b_{x_1} \\ b_{y_1} \\ b_{w_1} \\ b_{h_1} \\ P(Object)_2 \\ b_{x_2} \\ b_{y_2} \\ b_{w_2} \\ b_{h_2} \\ c_1 \end{bmatrix} \quad (4)$$

Hasil dari anotasi tersebut adalah data yang terdapat informasi letak kotak pembatas dan labelnya dalam bentuk .txt. Pada .txt file terdapat baris file yang memiliki format `<object-class> <x_center> <y_center> <width> <height>`, dimana pada `<object-class>` merupakan bilangan bulat yang menyatakan kelas objek, `<x_center>` dan `<y_center>` adalah koordinat pusat persergi kotak pembatas,

`<width>` dan `<height>` adalah nilai float relatif terhadap dimensi gambar.

Data Augmentation

Data Augmentation adalah teknik yang memperkenalkan sampel baru klasifikasi dengan menerapkan transformasi pada sampel nyata. Augmentasi data dalam analisis data adalah teknik yang digunakan untuk menambah jumlah data dengan menambahkan salinan yang sedikit dimodifikasi dari data yang sudah ada atau data sintesis yang baru dibuat dari data yang ada.



Sumber : Penulis (2022)

Gambar 2. Data Augmentation

Proses data augmentation citra menggunakan *Roboflow* untuk membudahkan proses data agumentation dan penyimpanan data citra sebagai API agar dataset citra dapat di panggil di berbagai komputer. Untuk memodifikasi data nyata dengan memutar gambar beberapa derajat, memperbesar sedikit atau membalik gambar dan mozaik merupakan yang dibuat dengan menggabungkan banyak kepingan foto kombinasi dari dataset. Teknik ini menciptakan variasi gambar yang ada di dataset dan dapat dilihat sebagai alternatif yang murah untuk menghasilkan lebih banyak data berlabel. Ini bertindak sebagai pengatur dan membantu mengurangi overfitting saat melatih model. Hasil dari data augmentation dilakukan multiplikasi dari *Dataset* training set dari 288 gambar citra multiplikasi sebanyak 12 kali menjadi 3430 gambar citra untuk data validasi dan data test tidak di lakukan data *Augmentation*.

Environment

Penelitian ini menggunakan *Environment* spesifikasi komputer dengan OS Ubuntu 22.04 LTS, Prosesor Intel(R) I9 12900K, RAM 64 GB, dan GPU Tesla P100(16GB).

Fase Training

Arsitektur YOLOv5 dibangun berdasarkan teori dan dirilis melalui repositori di GitHub. Seperti yang disebutkan, *Ultralytic* membangun

YOLOv5 pada kerangka kerja *PyTorch*, salah satu kerangka kerja paling populer di komunitas AI.

Layer pertama dalam *CNN* adalah input layer, pada input layer menerima masukan citra gambar dengan tiga atribut, yaitu panjang citra (*pixel*), lebar citra (*pixel*) dan *channel* warna (*RGB* atau *grayscale*). Karena penelitian ini adalah supervised learning, maka pada *input* citra pelatihan juga menggunakan label.

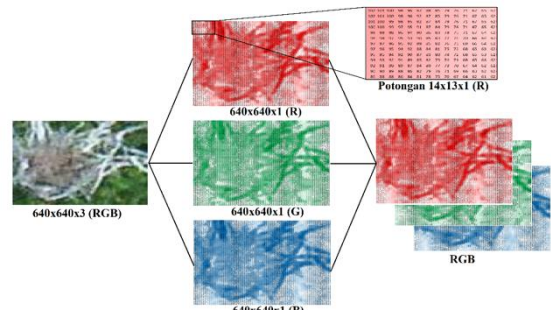


Sumber : Penulis (2022)
 Gambar 3. Citra hutan terdapat sarang Orang Utan

$$y = \begin{bmatrix} P(Object)_1 \\ b_{x_1} \\ b_{y_1} \\ b_{w_1} \\ b_{h_1} \\ P(Object)_2 \\ b_{x_2} \\ b_{y_2} \\ b_{w_2} \\ b_{h_2} \\ c_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.596 \\ 0.308 \\ 0.0123 \\ 0.016 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Setiap kotak pembatas terdiri dari 5 prediksi: x , y , w , h , dan nilai keyakinan p . Koordinat (x, y) mewakili pusat kotak relatif terhadap batas sel petak. Lebar (w) dan tinggi (h) diprediksi relatif terhadap keseluruhan gambar. Akhirnya prediksi nilai keyakinan menyatakan *IoU* antara kotak yang diprediksi dan kotak *ground truth*. Tahapan pertama dalam melakukan pelatihan terhadap model adalah memasukkan data citra bunga ke input layer, pada layer ini citra gambar dikonversi kedalam matriks tiga dimensi, dengan ukuran panjang \times lebar \times 3 channel RGB (Red, Green, Blue). Pada penelitian ini nilai RGB pada setiap

pixel dinormalisasi menjadi rentang 0-1 untuk mempermudah proses komputasi dengan cara membagi setiap nilai RGB setiap pixel dengan 255.



Sumber : Penulis (2022)

Gambar 4. Citra sarang Orang Utan dalam RGB channel (0-225)

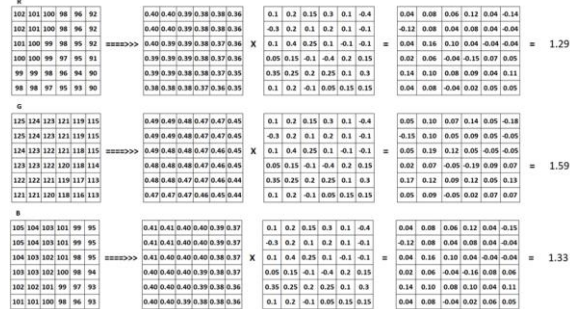
Data besaran nilai RGB channel pada setiap pixel kemudian diproses di lapisan konvusi (*convolution layer*). Fungsi lapisan ini adalah untuk mengekstrasi fitur- fitur (*feature map*) yang ada pada citra dengan menggunakan filter. Sebagai contoh, pada Gambar adalah input citra pada *channel*. Angka-angka yang tersebut mewakili besaran nilai pada citra gambar tersebut (0-255). Untuk mendapatkan fitur dari citra maka citra tersebut maka diperlukan proses konvolusi dengan *filter*, *filter* sendiri merupakan kumpulan nilai (*weight*) yang diinisiasi secara acak. Contoh nilai filter terdapat pada Gambar 5 di bawah ini.

0.1	0.2	0.15	0.3	0.1	-0.4
-0.3	0.2	0.1	0.2	0.1	-0.1
0.1	0.4	0.25	0.1	-0.1	-0.1
0.05	0.15	-0.1	-0.4	0.2	0.15
0.35	0.25	0.2	0.25	0.1	0.3
0.1	0.2	-0.1	0.05	0.15	0.15

$$S(i,j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n)$$

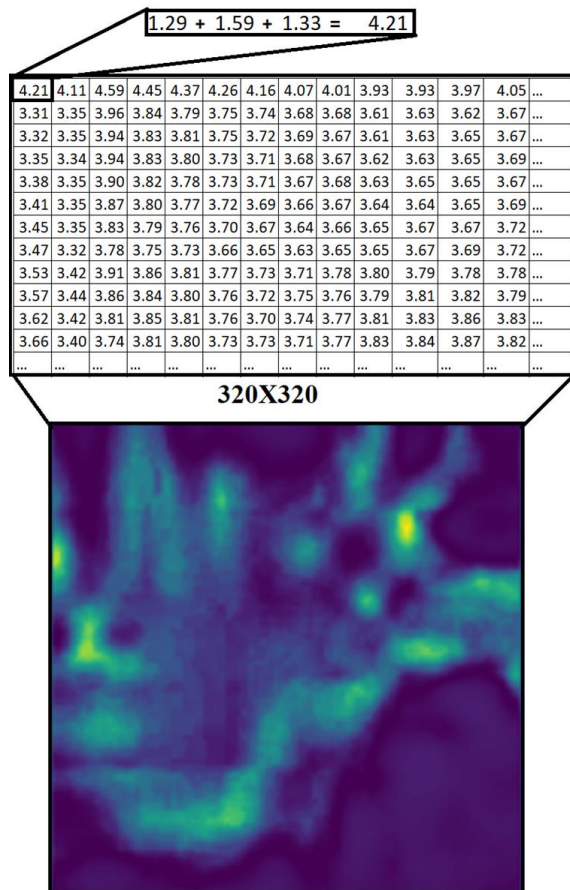
$S(i,j)$ adalah nilai output dari proses konvolusi pada *pixel* (i,j) dari gambar input. $I(i,j)$ adalah nilai pixel pada posisi (i,j) dari gambar input. $K(m,n)$ adalah nilai *pixel* pada posisi (m,n) dari kernel (atau matriks konvolusi) yang digunakan dalam proses konvolusi. $\sum_m \sum_n$ adalah notasi sigma, yang berarti penjumlahan dari setiap nilai m dan n . $K * I$ adalah operator konvolusi yang digunakan untuk menentukan nilai output $S(i,j)$ dari proses konvolusi. $S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m,j-n)K(m,n)$ adalah formula untuk menentukan nilai output $S(i,j)$ dari proses konvolusi yang digunakan untuk menentukan nilai output dari setiap pixel dari gambar input dengan mengalikan nilai pixel dari gambar input dengan nilai pixel dari kernel yang digunakan.

Ini berarti proses konvolusi akan menentukan output dari setiap pixel pada gambar input dengan mengalikan nilai pixel dari gambar input dengan nilai pixel dari kernel yang digunakan, dan menjumlahkan hasilnya, ini akan digunakan untuk menentukan nilai pixel pada output gambar.



$$S(i,j) = (K * I)(i, j)$$

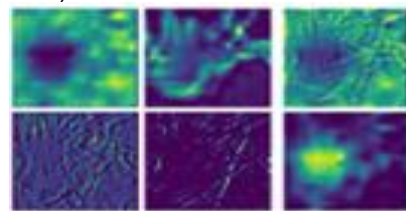
$S(i, j)$ adalah hasil dari perkalian matriks K dengan matriks I (matriks identitas). K adalah matriks konstanta yang digunakan dalam perhitungan, sedangkan I adalah matriks identitas yang memiliki diagonal utama berisi 1 dan elemen lainnya berisi 0. Indeks i dan j dalam $S(i, j)$ menunjukkan posisi elemen di dalam matriks hasil perkalian.



Sumber : Penulis (2022)

Gambar 5. Proses Convolution

Pada setelah lapisan konvolusional terjadi proses ekstraksi fitur untuk menghasilkan peta fitur (*feature map*) pada data citra, untuk mendapatkan mengenal pola pada setiap citra. Proses ekstraksi fitur dilakukan dengan *filter* pada lapisan konvolusional dengan besaran sesuai dengan besaran *kernel*. Visualisasi terhadap nilai *filter* pada lapisan konvolusi pertama yang digunakan pada data sarang urang utan terdapat pada (Gambar 5) *Filter* tersebut kemudian dikonvolusi dengan data input sehingga menghasilkan peta fitur yang dapat mendeteksi tepian (*edge*) yang membentuk pola (Gambar 6), proses konvolusi dan *pooling* yang dilakukan terus menerus akan membentuk pola yang lebih detail (Gambar 6).



Sumber : Penulis (2022)

Gambar 6. Hasil Convolution

Untuk dapat mengenal objek dalam citra diperlukan pemisahan objek dengan latar belakang pada objek. Pada penelitian ini activation *function ReLU* digunakan untuk menentukan aktif tidaknya neuron pada neural networks sehingga hanya neuron yang berhubungan dengan objek bunga saja yang dipilih dan selain itu tujuan dari fungsi aktivasi adalah untuk menambahkan properti *non-linear* ke fungsi, yang merupakan *neural network*.

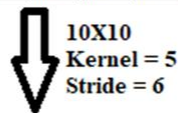
1.09	0.69	-0.72	0.76	0.77	0.95	1.06	1.14	1.20	1.25	1.38	1.22	...
1.95	1.85	1.97	1.85	1.93	2.00	2.01	1.98	2.02	2.13	1.99	2.18	...
1.95	1.87	1.78	1.79	1.92	1.89	1.80	-1.95	2.00	2.10	2.11	2.18	...
1.95	1.98	1.84	1.90	-1.84	1.82	1.83	1.83	2.02	2.05	2.05	2.05	...
1.96	-1.88	-1.68	-1.79	1.78	1.97	1.97	2.02	1.98	2.11	2.02	2.15	...
1.96	1.84	-1.78	1.79	-1.82	1.78	1.84	-1.96	-2.09	2.08	2.23	2.12	...
1.96	1.87	-1.68	1.77	-1.69	1.73	-1.85	1.98	2.09	2.14	2.08	2.19	...
1.96	1.87	1.70	1.79	1.83	2.01	1.99	-1.89	1.96	2.12	2.08	2.20	...
1.96	1.91	1.74	1.80	1.75	1.88	-1.86	2.03	-1.95	2.05	2.07	-2.05	...
1.96	1.87	-1.61	-1.62	-1.84	-1.88	-1.84	2.11	-2.06	-2.04	-2.17	-2.15	...
1.96	1.99	1.58	1.70	1.83	1.92	2.04	1.97	-2.06	-2.12	-2.08	-2.21	...
1.95	1.99	1.75	1.83	1.75	1.77	2.07	-1.93	1.94	-1.84	-1.98	-2.03	...
...

$$\text{Relu } f(z) = \max(0, z)$$

1.09	0.69	0.00	0.76	0.77	0.95	1.06	1.14	1.20	1.25	1.38	1.22	...
1.95	1.85	1.97	1.85	1.93	2.00	2.01	1.98	2.02	2.13	1.99	2.18	...
1.95	1.87	1.78	1.79	1.92	1.89	1.80	0.00	2.00	2.10	2.11	2.18	...
1.95	1.98	1.84	1.90	0.00	1.82	1.83	1.83	2.02	2.05	2.05	2.05	...
1.96	0.00	0.00	0.00	1.78	1.97	1.97	2.02	1.98	2.11	2.02	2.15	...
1.96	1.84	0.00	1.79	0.00	1.78	1.84	0.00	0.00	2.08	2.23	2.12	...
1.96	1.87	0.00	1.77	0.00	1.73	0.00	1.98	2.09	2.14	2.08	2.19	...
1.96	1.87	1.70	1.79	1.83	2.01	1.99	0.00	1.96	2.12	2.08	2.20	...
1.96	1.91	1.74	1.80	1.75	1.88	0.00	2.03	0.00	2.05	2.07	0.00	...
1.96	1.87	0.00	0.00	0.00	0.00	0.00	2.11	0.00	0.00	0.00	0.00	...
1.96	1.99	1.58	1.70	1.83	1.92	2.04	1.97	0.00	0.00	0.00	0.00	...
1.95	1.99	1.75	1.83	1.75	1.77	2.07	0.00	1.94	0.00	0.00	0.00	...
...

Pooling layer berfungsi untuk mengurangi ukuran spasial dari citra dan mengurangi jumlah parameter dan perhitungan dalam neural network. *Pooling layer* beroperasi pada setiap fitur secara independen. Pada lapisan ini menerima input dari hasil peta fitur (*feature map*) dari hasil proses konvolusi pada convolution layer. Penelitian ini menggunakan max pooling ukuran pooling 5x5 dengan stride 6 sehingga ukuran citra yang dihasilkan 50% ukuran lebih kecil.

3.69	3.38	3.68	3.76	3.75	3.73	3.69	3.70	3.72	3.79	...
3.70	3.51	3.83	3.89	3.89	3.84	3.87	3.89	3.92	3.96	...
3.71	3.55	3.88	3.92	3.91	3.89	3.88	3.94	3.97	4.00	...
3.71	3.53	3.86	3.90	3.90	3.88	3.90	3.94	3.98	4.00	...
3.70	3.52	3.86	3.92	3.92	3.93	3.92	3.92	3.97	4.00	...
3.65	3.53	3.89	3.94	3.91	3.92	3.96	3.96	3.96	4.02	...
3.60	3.52	3.93	3.96	3.96	3.95	3.96	3.99	3.97	3.97	...
3.54	3.58	4.06	4.05	4.04	4.07	4.09	4.12	4.10	4.12	...
3.47	3.59	4.11	4.08	4.08	4.07	4.08	4.10	4.14	4.12	...
3.39	3.61	4.18	4.09	4.08	4.09	4.07	4.08	4.11	4.13	...
...



3.92	4.00	...
4.18	4.14	...
...

Proses ini di laukan secara berulang-ulang dan terus menerus untuk mengekstrak peta fitur (*feature map*) dari Dataset.

Hasil Training model YOLOv5

Setelah selesai pelatihan, testing dan validasi maka yang terakhir adalah mengkalkulasikan hasil akurasi dari keseluruhan epoch. Berikut tabel 1 ini adalah hasil dari Epoch terbaik keseluruhan.

Tabel 1. Hasil Training model

Best Epoch	
Epoch	217
Precision	0.973
Recall	0.949
mAP_0.5	0.969
mAP_0.5:0.95	0.630

Sumber: Penulis (2022)

Dari tabel 1 Dengan dataset yang berisi 3430 gambar train dan 85 gambar validasi, model membutuhkan waktu 16 Menit untuk menyelesaikan satu *epoch*, dan 58 Jam untuk 217 *epoch* menghasilkan *Precision* 0.973, *Recall* 0.949 *mAP*_0.5 0.969 *mAP*_0.5:0.95 0.630. Ini membuktikan bahwa, hanya dengan arsitektur asli YOLOv5, modelnya tidak hanya cepat, tetapi akurasinya juga tinggi.

Untuk mengevaluasi kinerja algoritma pendeteksian objek dengan *Precision* saja. Juga, tidak tepat untuk mengevaluasi kinerja dengan *Recall* saja. *Precision* dan *Recall* selalu keluar sebagai nilai antara 0 dan 1. Jika *Precision* tinggi, *Recall* cenderung rendah, dan jika *Precision* rendah, *Recall* cenderung tinggi. Oleh karena itu, perlu untuk mengevaluasi kinerja algoritma dengan mensintesis dua nilai, dan dengan demikian diperoleh kurva *Precision-Recall* dan *mAP*.

4. Pengujian

Pada tahap pengujian sistem akan dilakukan evaluasi kinerja pendeteksian citra untuk keperluan analisis tingkat akurasi dan bobot terlatih dapat digunakan untuk mengidentifikasi sarang orangan. Jika keberadaan sarang orangan terdeteksi, kotak pembatas digambar untuk membungkus objek dan menampilkan probabilitas bahwa objek tersebut adalah sarang orangan. Metode melakukan deteksi objek dengan bobot terlatih mirip dengan melatih model. Menggunakan perintah yang file *detect.py* akan dikompilasi, dan itu membangun kembali arsitektur yang digunakan dalam pelatihan.



Sumber : Penulis (2022)

Gambar 7. Pengujian Deteksi Sara Orang Utan

Pada pengujian ini dilakukakan dengan menggunakan bobot terlatih terbaik dari hasil latih dimana gambar yang tidak pernah di lihat oleh bobot terlatih, pelabelan manual dari *dataset test* akan di sandingkan dengan hasil prediksi dari bobot terlatih

Tabel 2. Hasil Pengujian 10 Citra Gambar

No	File_name	Jumlah Sarang Expert Label	Jumlah Sarang prediksi YOLOv5	%
1	DSC00084_geotag.jpg	1	1	100
2	SC00085_geotag.jpg	2	2	100
3	DSC00089_geotag.jpg	1	1	100
4	DSC00169_geotag.jpg	3	3	100
5	DSC00177_geotag.jpg	1	1	100
6	DSC07644_geotag.jpg	1	1	100
7	DSC07660_geotag.jpg	1	1	100
8	DSC07670_geotag.jpg	1	1	100
9	DSC07672_geotag.jpg	3	3	100
10	DSC07774_geotag.JPG.jpg	1	1	100

Sumber: Penulis (2022)

Dari pengujian yang di lakukan pada tabel 2 untuk menguji ulang metode algoritma *You Only Look Once (YOLO)* dari *Best Training Weight Model PyTorch* di lakukan pengujian dari 10 data citra test dari pengujian di dapatkan bahwa model dapat mendeteksi jumlah sarang orang utan dengan rata-rata akurasi 99,9% dan YOLOv5 mampu untuk melakukan deteksi sarang orang utan pada cita gambar hutan.

4. Kesimpulan

Implementasi *You Only Look Once (YOLO)* untuk deteksi Sarang Orang Utan yaitu dengan meng-input citra gambar sebanyak 1970 citra gambar dan 414 gambar yang memiliki label Sarang Orang Utan berukuran 1920x1080 pixel, skenario perbandingan *dataset train* sebesar 79% dan *dataset test* sebesar 21% menggunakan aktivasi *ReLU*, 270 layers, 7022326 parameters, 7022326 gradients, 15.8 GFLOPs dan jumlah epoch 217 dimana menggunakan jenis gambar berwarna (RGB). Hasil training model dengan dataset dari hasil dataset training set dari 288 gambar citra di multiplikasi sebanyak 12 kali menjadi 3430 gambar citra *train* dan 85 gambar validasi, model membutuhkan waktu 16 Menit untuk menyelesaikan satu *epoch*, dan 58 Jam untuk 217 *epoch* menghasilkan *Precision* 0.973,

Recall 0.949 *mAP_0.5* 0.969 dan *mAP_0.5:0.95* 0.630 Semakin tinggi nilai *mAP* maka pendeteksi objek semakin akurat. Untuk menguji ulang metode algoritma *You Only Look Once (YOLO)* dari *best training weight model PyTorch* di lakukan pengujian dari 10 data citra test dari pengujian di dapatkan bahwa model dapat mendeteksi jumlah sarang orang utan dengan rata-rata akurasi 99,9%.

Referensi

- Anne K. Schütz , Verena Schöler , E. Tobias Krause , Mareike Fischer (2021) Application of YOLOv4 for Detection and Motion Monitoring of Red Foxes.
- Deng, L., & Yu, D. (2013). Deep learning: Methods and applications. In Foundations and Trends in Signal Processing. <https://doi.org/10.1561/20000000039>
- Eka Putra, W. S. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. Jurnal Teknik ITS, 5(1). <https://doi.org/10.12962/j23373539.v5i1.15696>
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics. <https://doi.org/10.1007/BF00344251>
- Gunawan, I. A. (2014). Buku Statistik Taman Nasional Sebangau. Palangkaraya: Balai Taman Nasional.
- Harahap, M., Elfrida, J., Agusman, P., Rafael, M., Abram, R., & Andrianto, K. (2019). Sistem Cerdas Pemantauan Arus Lalu Lintas Dengan YOLO (You Only Look Once v3). 367–376.
- Junita Sri Wisna, Tekad Matulatan, Nurul Hayaty (2019) Deteksi Kendaraan Secara Real-Time Menggunakan Metode YOLO Berbasis Android.
- LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vision. ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems. <https://doi.org/10.1109/ISCAS.2010.5537907>
- Liu, G. (2017), Real-time object detection for autonomous driving-based on deep learning, PhD thesis.
- Mahmudi, A., & Rusda, M. T. (2014). Deteksi Senjata Tajam Dengan Metode Haar Cascade Classifier Menggunakan Teknologi Sms Gateway. Matics, 1(1), 27–30. <https://doi.org/10.18860/mat.v1i1.2646>

- Mauricio Menegaz. (2018). Understanding YOLO – Hacker Noon. Hackernoon.
- Munir, R. (2004). Pengantar Pengolahan Citra (pp. 1–14).
- O’Shea, K. dan Nash, R. (2015), ‘An introduction to convolutional neural networks’, arXiv preprint arXiv:1511.08458 .
- Olmos, R., Tabik, S., & Herrera, F. (2018). Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*, 275(July 2020), 66–72. <https://doi.org/10.1016/j.neucom.2017.05.012>
- Pramestya, Ravy Hayu (2018) Deteksi dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode YOLO berbasis Citra Digital. Masters thesis, Institut Teknologi Sepuluh Nopember.
- Republik indonesia. (1982, januari). Undang-Undang Republik Indonesia. Nomor 4 Tahun 1982 Tentang ketentuan-ketentuan pokok pengelolaan lingkungan hidup.
- Richardson Santiago Teles de Menezes, John Victor Alves Luiz, Aron Miranda Henriques-Alves (2019) Mice Tracking Using The YOLO Algorithm.
- Setyawan, A. (2002). Biodiversitas Genetik, Spesies dan Ekosistem Mangrove di Jawa. Surakarta: Kelompok kerja biodiversitas jurusan biologi fakultas matematika dan pengetahuan alam.
- Van Schaik, C. P. 2006. Diantara Orangutan Kera Merah dan Bangkitnya Kebudayaan Manusia. Cetakan Pertama. Yayasan Penyelamatan Orangutan Borneo: Jakarta.