

Deteksi Karakter Berbasis YOLOv5 dan U-Net untuk Aksara Nusantara

Agi Prasetiadi¹, Julian Saputra², Iqsyahiro Kresna^{A3}, Imada Ramadhanti⁴

^{1,2,3,4}Fakultas Teknik Informatika, Institut Teknologi Telkom Purwokerto, Indonesia

Info Artikel

Riwayat artikel:

Diterima 05 September 2023

Direvisi 09 November 2023

Diterima 12 November 2023

Diterbitkan 28 Desember 2023

Kata kunci:

Bounding Box
Deteksi Karakter
Aksara Nusantara
YOLO
U-Net

ABSTRAK

Indonesia memiliki beragam aksara asli, yang disebut aksara Nusantara, yang meliputi aksara Bali, Batak, Bugis, Jawa, Kawi, Kerinci, Lampung, Pallawa, Rejang, dan Sunda. Namun, teknik pendeteksian karakter yang ada saat ini sebagian besar melayani aksara Latin atau Cina. Sebagai kelanjutan dari penelitian kami sebelumnya, yang berkonsentrasi pada klasifikasi jenis aksara dan pengenalan karakter dalam sistem aksara Nusantara, penelitian ini mengembangkan penelitian kami dengan mengintegrasikan teknik pendeteksian objek, menggunakan model YOLOv5, dan meningkatkan kinerja melalui penggabungan model U-Net untuk memfasilitasi penentuan lokasi karakter aksara Nusantara yang mendasar dalam gambar dokumen masukan. Selanjutnya, penelitian kami mempelajari pengaturan ulang posisi karakter ini agar selaras dengan gaya khas aksara Nusantara. Hasil eksperimen menunjukkan kinerja YOLOv5, menghasilkan tingkat kehilangan sekitar 0,05 dalam deteksi lokasi karakter. Sementara itu, model U-Net menunjukkan akurasi mulai dari 75% hingga 90% untuk memprediksi wilayah karakter. Meskipun YOLOv5 mungkin tidak dapat mendeteksi semua aksara Nusantara dengan sempurna, model U-Net secara signifikan meningkatkan tingkat deteksi sebesar 1,2%.

Penulis Korespondensi:

Agi Prasetiadi,

Program Studi Teknik Informatika, Fakultas Informatika, Institut Teknologi Telkom

Purwokerto Jl. Panjaitan No. 128, Purwokerto, Indonesia. 53147

Email: agi@ittelkom-pwt.ac.id

1. PENDAHULUAN

Belum ada model pembelajaran mendalam Optical Character Recognition (OCR) yang komprehensif untuk mendeteksi aksara Nusantara. Berdasarkan data Google Scholar, makalah penelitian yang terkait dengan pengenalan dan pendeteksian aksara Nusantara, termasuk Jawa, Batak, Bali, Pallawa, Lontara, Kawi, Rejang, Sunda, Lampung, dan Kerinci, jumlahnya bervariasi, dengan jumlah masing-masing 800-an, 60-an, 400-an, 30-an, 90-an, 70-an, 10-an, 150-an, 100-an, dan 10-an. Bahasa Jawa, Bali, dan Sunda telah mendapatkan pengakuan yang cukup besar, sementara bahasa Pallawa dan Rejang masih kurang terwakili secara signifikan, dan tidak memiliki model pengenalan atau pendeteksian yang terimplementasi. Selain itu, di antara penelitian-penelitian tersebut, topik OCR mendapat perhatian terutama untuk aksara Jawa dan Bali, dengan sekitar 10 makalah penelitian yang tersedia, yang hanya menawarkan kemampuan deteksi parsial.

OCR melibatkan pengenalan teks dalam dokumen atau gambar dan mengubahnya menjadi teks yang dapat dibaca oleh mesin [1]. Hal ini memfasilitasi digitalisasi konten dokumen dan meningkatkan efisiensi operasional [2]. Bisnis banyak menggunakan OCR untuk mengekstrak informasi dari sumber media cetak [3]. Namun, aksara Nusantara, yang mencakup kumpulan aksara tradisional asli Indonesia, telah menghadapi kepunahan karena perkembangan kontemporer [4]. Untuk menangkalkan penurunan ini dan melindungi naskah-naskah yang tak ternilai ini, kami melakukan penelitian dengan tujuan mengembangkan program yang mampu membaca aksara Nusantara.

sebuah program OCR khusus untuk aksara Nusantara. Penelitian ini menggunakan set data yang terdiri dari 10 jenis aksara.

Pada dasarnya, ada beberapa metode klasifikasi gambar untuk membaca karakter, termasuk Metode Pencocokan Segmen Terstruktur [5], Analisis Morfologi [6], Evaluator berbasis diskriminan pada fitur yang diekstraksi pada gambar biner yang dinormalisasi [7], Jaringan Syaraf Tiruan [8], dan Jaringan Syaraf Tiruan yang Dapat Dipisahkan Secara Depthwise [9]. Namun, fase mendasar dalam deteksi lokasi karakter melibatkan penerapan teknik bounding box, yang digunakan untuk mengisolasi dan kemudian memproses masing-masing karakter untuk klasifikasi. Penelitian sebelumnya telah mengeksplorasi berbagai algoritma untuk tujuan ini, termasuk Convolutional Neural Networks (CNN), Long-Short Term Memory (LSTM), You Only Look Once (YOLO). Seluk-beluk setiap algoritma akan diuraikan sebagai berikut.

YOLOv4 telah digunakan untuk melatih model CNN untuk mengenali karakter Arab tercetak, mencapai akurasi 82,4% [10]. Peneliti lain mengeksplorasi metode untuk mengembangkan sistem OCR untuk Naskah Sansekerta, menggunakan ekstraksi fitur konvensional, metode heuristik, dan pendekatan pembelajaran mesin seperti CNN, LSTM, atau LSTM dua arah [11]. Sistem OCR untuk Aksara Jawa dibuat menggunakan Projection Profile Segmentation dan Nearest Centroid Classifier, mencapai tingkat keberhasilan 93,88% untuk segmentasi baris dan 73,59% untuk segmentasi karakter, dengan akurasi klasifikasi 60,6% [12]. Faster Region-based Convolutional Neural Network (Faster R-CNN) juga telah digunakan dalam pendeteksian aksara Jawa, dengan tingkat akurasi berkisar antara 41,67% hingga 96,31% [13]. YOLOv4 telah menunjukkan tingkat deteksi yang mengesankan, yaitu hampir 99,55% untuk aksara Bali pada naskah lontar tradisional [14]. Terlepas dari berbagai versi YOLO, YOLOv5 mengungguli versi lainnya dalam lingkungan yang menantang seperti skenario bawah air [15]. Akhirnya, di luar deteksi lokasi karakter, U-Net dengan perhatian multi-tahap telah terbukti efektif dalam menentukan wilayah teks pada dokumen seperti faktur, tanda terima, dan transfer bank [16].

Dalam kasus kami, untuk membangun sistem OCR yang komprehensif untuk aksara Nusantara, sangat penting untuk membangun tiga model yang berbeda: model klasifikasi jenis aksara, model pengenalan karakter yang disesuaikan dengan masing-masing aksara Nusantara, dan model untuk deteksi lokasi karakter dalam aksara Nusantara. Dua model pertama telah dikembangkan dalam penelitian sebelumnya, seperti yang didokumentasikan dalam [17]. Namun, penelitian ini berpusat pada model deteksi karakter. Pendekatan yang digunakan dalam makalah ini terutama berfokus pada deteksi karakter dasar untuk memfasilitasi prosedur prapemrosesan dan pelatihan yang layak. Model yang dirancang untuk pelatihan dataset terdiri dari perpaduan antara model YOLOv5 dan U-Net. YOLOv5 berfungsi sebagai fondasi untuk deteksi aksara Nusantara karena ketangguhannya, sementara U-Net membantu dengan menunjukkan area yang menarik dengan menggunakan topeng, melengkapi YOLOv5 jika terjadi kesalahan deteksi karakter. Penilaian kinerja model-model ini akan dilakukan berdasarkan metrik kehilangan dan akurasi masing-masing.

2. METODE

2.1. Aksara Nusantara

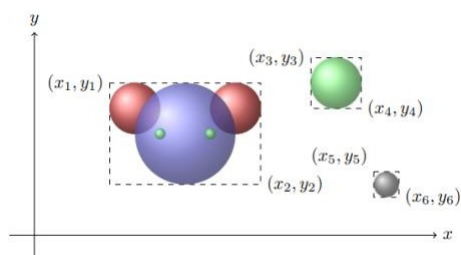
Indonesia memiliki sejarah aksara Nusantara yang kaya, yang telah digunakan sejak zaman kuno untuk ekspresi bahasa [18]. Aksara Pallawa, yang dipengaruhi oleh bahasa Sansekerta, merupakan aksara yang paling awal digunakan di Indonesia dan mempengaruhi daerah-daerah seperti Sumatra, Jawa, Bugis, dan Bali. Berbagai aksara Nusantara, termasuk aksara Jawa, Bali, Lontara (Bugis), Rejang, Lampung, Batak, Pallawa, Kawi, Sunda, dan Kerinci, memiliki signifikansi budaya tetapi penggunaan praktisnya terbatas dalam kehidupan sehari-hari [19]. Sebagian besar aksara Nusantara termasuk dalam jenis Abugida, di mana konsonan dan vokal digabungkan sebagai aksara dasar [20]. Perilaku aksara ini disebut dengan rendering teks yang kompleks dalam konteks komputasi.

Ambil contoh huruf Jawa 'ha'. Ketika dikombinasikan dengan diakritik Swara, huruf ini dapat menghasilkan setidaknya 5 variasi. Ini dapat berubah menjadi 4 variasi ketika dikombinasikan dengan diakritik Panyigeg dan 4 variasi lainnya dengan Wyanjana. Hal ini menghasilkan 150 variasi untuk satu huruf 'ha' ($1 \times 6 \times 5 \times 5$). Ekstrapolasi kerumitan ini untuk semua huruf dasar dalam aksara Jawa akan menghasilkan ribuan atau bahkan puluhan ribu variasi aksara. Menghadapi jumlah yang besar ini, kami telah memilih pendekatan yang bekerja dengan huruf dasar sebelum modifikasi apa pun. Pendekatan ini bertujuan untuk pengenalan huruf yang lebih cepat, lebih efisien, dan lebih akurat.

2.2. Deteksi Objek

Algoritma pendeteksian objek digunakan dalam pendeteksian karakter untuk mengklasifikasikan gambar ke dalam objek-objek diskrit [21]. Segmentasi kotak pembatas menetapkan kotak persegi panjang imajiner untuk setiap objek yang diidentifikasi, yang secara dinamis berukuran sesuai dengan objek tertentu. Hal ini didefinisikan oleh satu set koordinat, yang dapat mencakup (x_1, y_1) dan (x_2, y_2) , atau koordinat tunggal (x_1, y_1) bersama dengan parameter lebar (w) dan tinggi (h) [22]. Penelitian ini berfokus pada deteksi lokasi karakter, karena tidak ada pustaka yang tersedia untuk

mendeteksi karakter Nusantara. Bounding Box digunakan untuk menentukan lokasi karakter di dalam gambar, seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Lokasi objek yang terdeteksi menggunakan koordinat x_1, y_1, x_2, y_2 .

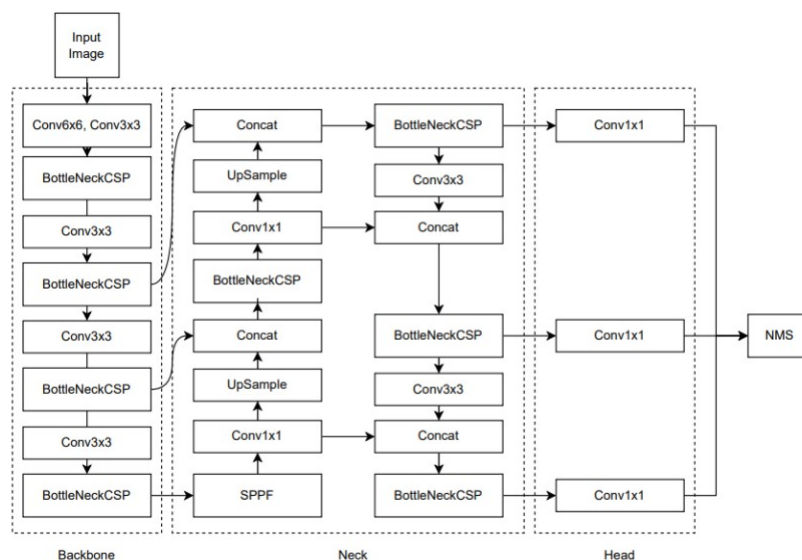


Gambar 2. Contoh aplikasi deteksi objek untuk bahasa Latin deteksi teks.

Teknik ini menggunakan input matriks 4 dimensi untuk analisis gambar, menghasilkan output yang terdiri dari dua pasangan koordinat yang mewakili lokasi objek. Berbagai arsitektur dapat digunakan, termasuk quadruplet CNN [23], Single Shot Detector (SSD) [24], YOLO [25], dan Mask RCNN [26]. Gambar 2 menunjukkan deteksi objek yang diterapkan pada deteksi teks. Penelitian kami menggabungkan dua model utama: YOLOv5 dan model U-Net untuk pelokalan karakter dalam aksara Nusantara.

YOLOv5, sebuah model CNN untuk deteksi objek [27] [28], menonjol karena kecepatannya yang mengesankan, mencapai FPS 62,5, lima kali lebih cepat daripada alternatif seperti YOLOv3 dan YOLOv4. Gambar 3 menunjukkan arsitektur YOLOv5, yang terdiri dari tiga komponen utama: Tulang Punggung, Leher, dan Kepala. Backbone mengekstrak fitur gambar yang penting untuk pemrosesan data yang efisien, menggunakan teknik BottleNeckCSP (Bottleneck Cross-Stage Partial Networks) untuk mengurangi jumlah saluran sambil mempertahankan informasi spasial [29].

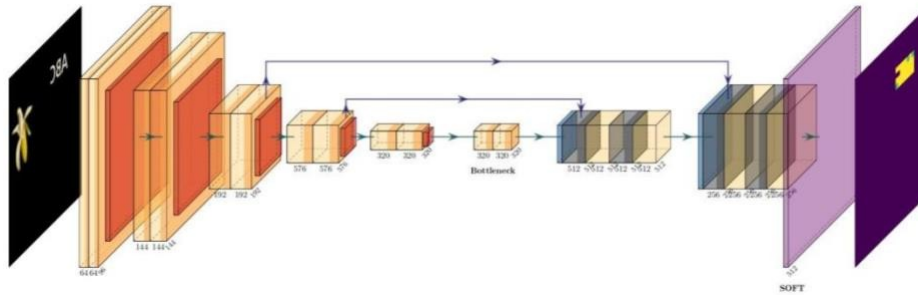
Neck menyempurnakan fitur, mengurangi noise dan meningkatkan aliran informasi melalui struktur seperti SPFF (Spatial Pyramid Fusion) dan CSP-PAN (Cross Stage Partial-Path Aggregation Network). SPFF menggabungkan fitur dari sel grid melalui MaxPooling, sementara CSP-PAN mendorong pertukaran informasi di seluruh tahap Backbone dan menangkap fitur dari berbagai tingkat abstraksi. Komponen Head terakhir menyempurnakan output dengan NMS (Non-Maximum Suppression), menghilangkan prediksi kotak pembatas duplikat dan memastikan kotak pembatas dengan tingkat



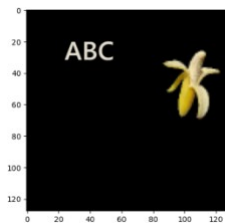
kepercayaan tertinggi untuk setiap objek yang terdeteksi [30].

Gambar 3. Arsitektur model YOLOv5.

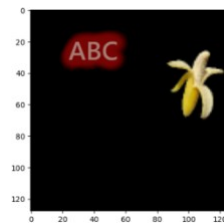
U-Net adalah model pembelajaran mendalam yang dirancang untuk segmentasi gambar biomedis tetapi digunakan untuk tugas segmentasi objek [31]. Gambar 4 desain arsitektur model U-Net yang digunakan dalam penelitian ini. Dalam penelitian ini, model U-Net digunakan untuk memprediksi lokasi teks di dalam sebuah gambar, meskipun pisang yang ditunjukkan pada Gambar 5a dan 5b. Model ini menggunakan lima tahap downsampling dan dua tahap upsampling untuk mengubah gambar input (128, 128) menjadi peta objek (32, 32). Fitur U-Net kemudian digunakan untuk mendeteksi lokasi semua aksara Nusantara, sehingga meningkatkan akurasi model YOLOv5 dalam mendeteksi karakter.



Gambar 4. Arsitektur model U-Net.



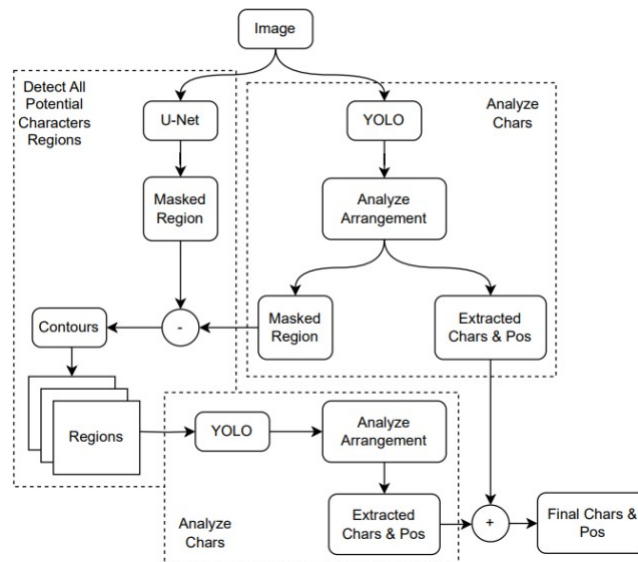
(a) Sebelum deteksi.



(b) Setelah deteksi dengan masker merah.

Gambar 5. Deteksi topeng objek U-Net untuk mendeteksi anjing.

Karena sifat inheren YOLOv5, tidak selalu menjamin bahwa semua karakter akan terdeteksi dalam satu kali pemeriksaan. Kami menggunakan strategi tambahan untuk mencari daerah karakter potensial menggunakan model U-Net untuk meningkatkan probabilitas menangkap karakter yang tersisa. Seperti yang diilustrasikan pada Gambar 6, fokus utama analisis karakter terletak pada sisi kanan, di mana model YOLOv5 pada awalnya memproses input. Secara bersamaan, sisi kiri aliran menunjukkan pemeriksaan untuk setiap wilayah karakter yang masuk akal dengan menggunakan model U-Net. Setiap daerah yang teridentifikasi kemudian dievaluasi kembali oleh model YOLOv5 untuk menguraikannya. Hasil dari analisis YOLOv5 dan U-Net kemudian digabungkan ke dalam daftar karakter yang komprehensif dan posisinya masing-masing.



Gambar 6. Pembacaan aksara Nusantara secara keseluruhan berdasarkan YOLOv5 dan U-Net.

Pada bagian 'Analisis Karakter', sebuah urutan berfokus pada penyempurnaan keluaran model YOLOv5 dengan mengatur ulang posisi karakter. Langkah ini sangat penting karena daftar awal posisi karakter yang disediakan oleh YOLOv5 sering kali perlu diurutkan. Pendekatan pengurutan kami sangat mudah: karakter awalnya diurutkan berdasarkan nilai sumbu y, yang mencerminkan ketinggiannya, untuk menyelaraskan dengan alur pembacaan tradisional aksara Nusantara dari atas ke bawah. Selain itu, kami bertujuan untuk menyusun aksara dari kiri ke kanan.

Namun, menyortir karakter dari kiri ke kanan menjadi sulit ketika titik tengahnya tidak sejajar dengan sumbu y yang sama. Untuk mengatasi hal ini, penelitian kami menggunakan algoritma pengelompokan KMeans, menggunakan data sumbu y untuk menyimpulkan jumlah kalimat atau garis yang mungkin ada di dalam

~~gambar [32]. Kami mulai dengan mempertimbangkan kalimat terkecil yang dapat dibuat dengan hanya menggunakan dua karakter. Dari sana, kita~~

bereksperimen dengan mengelompokkan koleksi posisi karakter, dimulai dari 2 cluster dan bergerak naik hingga jumlah total karakter dibagi 2. Label dihasilkan oleh algoritme pengelompokan kemudian dievaluasi menggunakan Silhouette Score, dengan skor tertinggi menunjukkan pengelompokan yang paling optimal. Silakan lihat Algoritma 1 untuk rincian algoritma yang lebih rinci.

Setelah koleksi dikelompokkan ke dalam klaster dan setiap klaster ditandai dengan tepat, langkah selanjutnya adalah menormalkan nilai sumbu y dari karakter. Selanjutnya, semua karakter yang memiliki label klaster yang sama, dikenakan proses penyortiran. Tepatnya, algoritme yang disebut **sortByCluster**(locations, clusters), mengatur pengaturan lokasi karakter dalam setiap baris, memastikan urutan dari kiri ke kanan yang mulus. Sisa dari algoritme ini didedikasikan untuk mengoptimalkan visualisasi karakter yang terdeteksi. Hal ini memerlukan penghitungan baris dan kolom yang diperlukan untuk menampilkan karakter dalam format yang mudah digunakan. Selain itu, semua posisi karakter diidentifikasi oleh model YOLOv5 dilambangkan sebagai peta YOLO.

Algorithm 1 Analyze Characters

```

1: function ANALYZECHARS(image)
2:    $R \leftarrow \text{YOLOModel}(\textit{image})$   $\triangleright$  Character locations
3:    $N \leftarrow \text{getCharCount}(R)$ 
4:   if  $N = 0$  then
5:      $\rightarrow \text{None}$ 
6:    $C \leftarrow \text{sortYX}(R)$   $\triangleright$  Sort by y-axis, then x-axis
7:   if  $N < \varepsilon$  then  $\triangleright$  We set  $\varepsilon$  as 6
8:      $L \leftarrow 1$   $\triangleright$  Consider as 1 line
9:   else
10:     $L \leftarrow \text{clusterLines}(C)$   $\triangleright$  Find lines/sentences
11:     $L \leftarrow \text{sortByCluster}(C, L)$   $\triangleright$  Group into lines
12:     $\text{mean\_len} \leftarrow \text{avgCharPerLine}(L)$ 
13:     $r, c \leftarrow \text{displayParams}(L, \text{mean\_len})$   $\triangleright$  Row and col
14:     $\rightarrow C, L, r, c, \text{mean\_len}$ 

```

Algorithm 2 Detect Other Characters

```

1: function DETECTOTHERCHARS(image)
2:    $I' \leftarrow \text{resize}(\textit{image})$   $\triangleright$  Resize and predict
3:    $Y \leftarrow \text{predictOtherChars}(I', M)$ 
4:    $Y \leftarrow \text{threshold}(Y)$ 
5:    $Co \leftarrow \text{findContours}(Y)$   $\triangleright$  Contours
6:    $P \leftarrow []$   $\triangleright$  Potential character regions
7:   for  $\text{cnt}$  in  $Co$  do
8:      $ce \leftarrow \text{centroid}(\text{cnt})$ 
9:     if  $\text{isPotentialChar}(ce, Y, \text{ref})$  then
10:       $x, y, w, h \leftarrow \text{bbox}(\text{cnt})$ 
11:       $P.append([x, y, w, h])$ 
12:   for  $x, y, w, h$  in  $P$  do  $\triangleright$  Reread in this region
13:      $T \leftarrow \text{crop}(I, x, y, w, h)$ 
14:      $C, L, r, c, \text{mean\_len} \leftarrow \text{AnalyzeChars}(T)$ 
15:     if  $R$  is None then
16:       continue
17:    $\text{showBoxes}(T, C, L, r, c, \text{mean\_len})$ 

```

Algoritma berikutnya menangani karakter yang tidak terdeteksi dan melibatkan penilaian sekunder (Algoritma 2). Kami menggunakan model U-Net, yang dilabeli sebagai **predictOtherChars**, untuk mengantisipasi wilayah karakter yang potensial. Setelah menghasilkan peta wilayah yang disarankan, kami mengekstrak kontur dari peta tersebut menggunakan fungsi **threshold** dan **findContours**, yang dilambangkan sebagai P untuk wilayah karakter potensial. Kontur-kontur ini memberikan posisi centroid sebagai titik referensi. Setiap titik pusat direferensikan dengan peta YOLO yang ada. Wilayah yang tidak ditemukan di peta YOLO menjalani evaluasi model YOLOv5 sekunder untuk memastikan tidak ada karakter yang terlewatkan dalam pemindaian awal. Karakter baru yang berhasil dideteksi ditambahkan ke koleksi utama karakter yang teridentifikasi dan posisinya masing-masing.

2.3. Pengumpulan Data

Dalam penelitian ini, kumpulan data yang luas telah dikumpulkan, mencakup 126 gambar yang mewakili 10 jenis Aksara Nusantara yang berbeda: Bali, Batak, Bugis, Jawa, Kawi, Kerinci, Lampung, Pallawa, Rejang, dan Sunda. Dataset ini menampilkan berbagai jenis dokumen dan foto, termasuk beberapa foto yang diambil dari jalanan umum. Sebagian dari dataset ini dikumpulkan dari Google Images, sementara sisanya diperoleh melalui tulisan tangan manual di atas kertas. Anotasi manual dilakukan pada gambar yang dikumpulkan untuk memastikan pelabelan yang tepat. Selanjutnya, dataset dikategorikan ke dalam tiga subset untuk pelatihan, validasi, dan pengujian. Sekitar 64% gambar dialokasikan untuk pelatihan, 17% untuk validasi, dan 19% untuk pengujian.

Proses anotasi menggunakan kotak pembatas atau persegi panjang untuk mendefinisikan lokasi objek, mengikuti metodologi yang diuraikan dalam [33]. Gambar 7 secara visual merepresentasikan proses ini. Mengingat fokus dataset pelatihan kami pada masing-masing huruf dasar, vokal, dan Panyigeg, kami secara khusus menganotasi komponen huruf 'ing' - huruf 'ha', vokal 'I', dan Panyigeg 'ng'. Pendekatan ini menargetkan pendeteksian huruf dasar individual daripada variasi dari mesin terbang 'ing'. Setelah klasifikasi huruf dasar selesai, langkah tambahan diperlukan untuk menghubungkan huruf-huruf dasar yang berbeda ini dan mendapatkan hasil pembacaan akhir.



Gambar 7. Semua anotasi skrip dengan mempertimbangkan posisi karakter dasar.

Selain itu, kami menerapkan teknik augmentasi data untuk mendiversifikasi dataset. Setiap gambar karakter menjalani tiga metode augmentasi yang berbeda, termasuk penyesuaian kontras, konversi skala abu-abu, rona, saturasi, kecerahan, eksposur, dan variasi noise untuk memperkaya variasi set data. Penambahan ini meningkatkan keragaman dan ketahanan dataset untuk pelatihan model deep learning selanjutnya.

2.4. Evaluasi

Ada 2 jenis evaluasi utama yang akan kami gunakan untuk mengukur kinerja model yang kami buat: kinerja loss untuk deteksi karakter dan kinerja akurasi untuk prediksi segmentasi karakter yang lebih luas. Penelitian-penelitian sebelumnya yang hanya menggunakan satu pendekatan deep learning mungkin sudah cukup untuk mengevaluasi aspek kinerja tertentu. Namun, dalam penelitian ini, dua teknik deep learning yang berbeda digunakan untuk tujuan yang berbeda: YOLOv5 untuk deteksi objek individu dan U-Net untuk memprediksi segmen di mana objek dapat didistribusikan. Pada evaluasi pertama, box, object, dan class loss akan digunakan untuk mengukur kinerja deteksi bounding box lokasi karakter. Berikut ini adalah penjelasan rinci dari setiap loss.

Box Loss mengevaluasi keakuratan prediksi koordinat dan dimensi kotak pembatas dibandingkan dengan kotak ground truth, yang sering kali dihitung menggunakan fungsi kehilangan regresi seperti MSE atau kehilangan L1 yang halus, sehingga membantu lokalisasi objek yang tepat. Object Loss, juga dikenal sebagai objectness loss, mengukur seberapa baik model memprediksi keberadaan objek di dalam kotak pembatas, sering kali menggunakan binary cross-entropy loss untuk melihat keberadaan objek secara akurat. Class Loss menilai keakuratan model dalam memprediksi kategori objek, mengukur keselarasan antara probabilitas kelas yang diprediksi dan label ground truth, biasanya dihitung dengan fungsi kehilangan klasifikasi seperti softmax cross-entropy atau log loss, meningkatkan klasifikasi objek yang benar.

Untuk evaluasi kedua, entropi silang kategorik jarang standar akan digunakan untuk mengukur wilayah prediksi lokasi karakter, di mana kerugian = $-\sum y_i \log y'_{(i)}$. Model prediksi wilayah karakter U-Net menghasilkan peta biner kecil yang mengindikasikan di mana karakter-karakter tersebut mungkin berada. Model ini membantu model utama YOLOv5 untuk mendeteksi ulang karakter yang terlewat dari pembacaan awal. Model ini bahkan mengembalikan wilayah karakter yang bukan milik aksara Nusantara.

3. HASIL DAN PEMBAHASAN

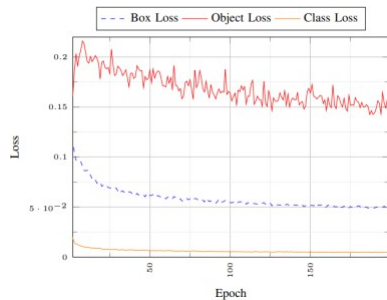
3.1. Kinerja Model YOLOv5

Gambar 8 mengilustrasikan analisis kinerja komprehensif dari model YOLOv5 selama proses pelatihan. Model ini dilatih menggunakan 64% dataset dengan konfigurasi 16 batch, yang berarti bahwa di setiap epoch, 1/16 dataset pelatihan diproses. Proses pelatihan berlangsung selama 200 epoch, sehingga seluruh dataset pelatihan dibaca sekitar 12,5 kali. Seiring berjalannya pelatihan, model menunjukkan penurunan yang konsisten dalam nilai kerugiannya dan tampaknya mendekati kondisi stabil. Secara khusus, dalam hal kehilangan kotak, model menunjukkan pemahaman yang bertahap tentang lokasi karakter, menghasilkan penurunan nilai kehilangan yang mulus menjadi sekitar 0,05. Ketika mengamati kehilangan objek, model menunjukkan beberapa fluktuasi, yang mencerminkan upaya untuk memastikan keberadaan objek di dalam kotak pembatas.

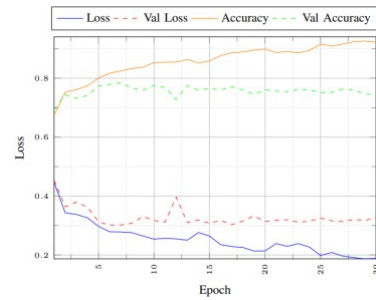
3.2. Kinerja Model U-Net

Gambar 9 menyajikan gambaran umum kinerja model U-Net selama fase pelatihan. Model ini menunjukkan penurunan yang stabil dalam nilai kerugian, yang mengindikasikan pelatihan yang efektif. Namun, pengamatan yang penting adalah kesenjangan yang relatif lebar antara kerugian pelatihan dan kerugian validasi, yang berjumlah sekitar 0,2. Perbedaan ini menunjukkan adanya overfitting, sebuah skenario di mana model mungkin terlalu disesuaikan dengan data pelatihan.

Dengan mengamati garis hijau dan merah pada grafik, terlihat jelas bahwa akurasi dan akurasi validasi menunjukkan tren yang meningkat dari waktu ke waktu. Hasil ini menandakan bahwa model ini secara progresif meningkatkan akurasi prediksinya. Khususnya, tingkat akurasi yang dicapai cukup tinggi, mencapai sekitar 0,9 di wilayah di mana terdapat karakter dan 0,75 dalam akurasi validasi. Hebatnya, model ini menunjukkan konvergensi yang cepat hanya dalam waktu 30 epoch.



Gambar 8. Performa model YOLOv5 pada Box, Object, dan Kehilangan Kelas selama pelatihan.



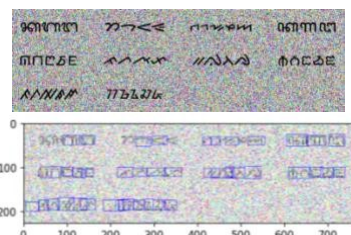
Gambar 9. Kinerja model U-Net Performa model U-Net pada wilayah karakter prediksi.

3.3. Hasil Percobaan

Bagian ini mengevaluasi kinerja model kami dalam membaca gambar dengan konten aksara Nusantara lengkap, prasasti, dan aksara campuran. Kemampuan membaca aksara Nusantara yang berbeda ditunjukkan pada Gambar 10. Gambar 10a menampilkan hasil pembacaan sepuluh aksara Nusantara yang telah didigitalisasi, yaitu aksara Bali, Batak, Kerinci (Incung), Jawa, Kawi, Lampung, Bugis (Lontara), Pallawa, Rejang, dan Sunda. Naskah-naskah ini bersumber dari aksaradinusantara.com. Pada Gambar 10b, aksara yang didigitalisasi terkena noise, sehingga mengakibatkan kesalahan deteksi 1 aksara Bali, 2 aksara Batak, dan 1 aksara Kerinci. Gambar 10c menampilkan koleksi foto penggunaan di dunia nyata yang menggambarkan 10 aksara Nusantara. Gambar 10d menampilkan aksara yang terdeteksi, dengan beberapa aksara yang masih belum terdeteksi. Gambar 10e menampilkan urutan akhir, setelah pemrosesan semua posisi aksara.



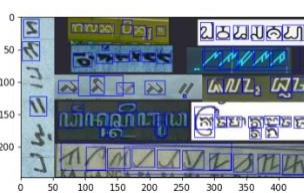
(a) Deteksi aksara Digital Nusantara sebelum (atas) dan sesudah (bawah).



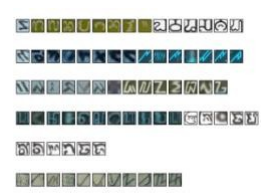
(b) Deteksi aksara Digital Nusantara dengan noise sebelum (atas) dan setelahnya (bawah).



(c) Semua foto naskah Nusantara yang diambil dari dunia nyata.



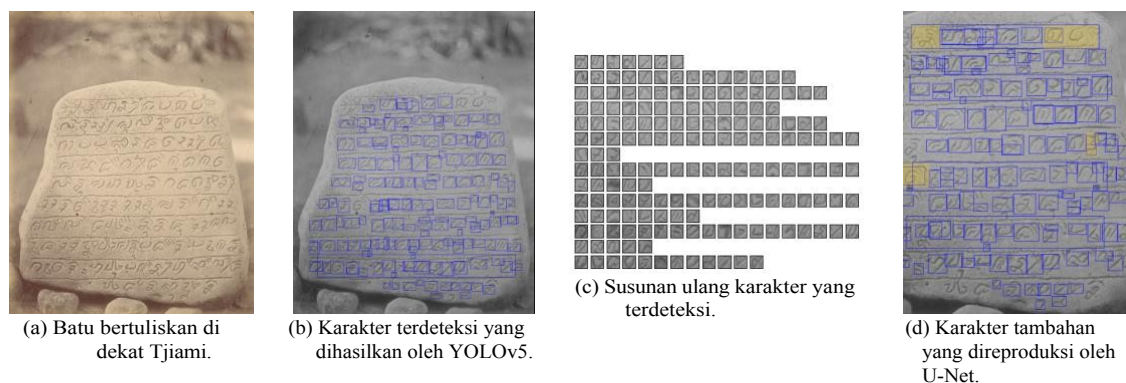
(d) Karakter terdeteksi yang dihasilkan oleh YOLOv5 dan U-Net.



(e) Karakter yang terdeteksi secara berurutan.

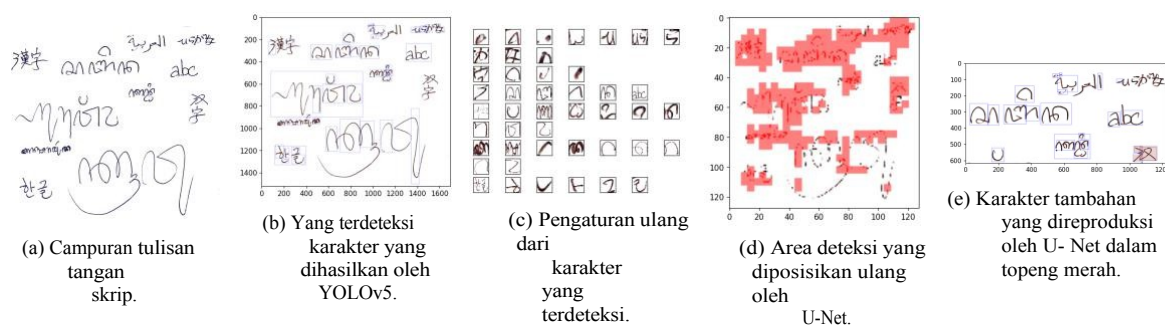
Gambar 10. Hasil deteksi aksara Nusantara Campuran hasil deteksi aksara Nusantara.

Gambar 11 mengilustrasikan deteksi karakter model kami pada sebuah foto historis dari sebuah batu bertuliskan di dekat Tjiami sebelum tahun 1900 [34]. Gambar tersebut pada awalnya dikonversi ke skala abu-abu untuk stabilitas (Gambar 11a). YOLOv5 mendeteksi sebagian besar karakter, dengan beberapa yang tidak terdeteksi pada awal kalimat (Gambar 11b). Posisi karakter kemudian diatur (Gambar 11c), dengan Algoritma 1 menentukan nomor baris dan Algoritma 2 mengoptimalkan penempatan karakter. Gambar 11d menunjukkan deteksi tambahan yang dibantu oleh daerah topeng U-Net, yang diperkenalkan kembali ke dalam YOLOv5, menyoroti karakter tambahan dalam kotak kuning. Secara khusus, U-Net melengkapi YOLOv5, memperkaya daftar karakter yang dikenali.



Gambar 11. Hasil deteksi karakter batu bertuliskan.

Gambar 12 menunjukkan kemampuan model kami untuk mendeteksi campuran aksara tulisan tangan, termasuk aksara Latin, Jawa, Arab, Hangeul, Cina Sederhana, Cina Tradisional, dan Hiragana. Pada awalnya, YOLOv5 berhasil mengidentifikasi semua aksara Nusantara dan sebagian besar aksara lainnya, kecuali aksara Tionghoa Sederhana (Gambar 12b). Gambar 12c menampilkan posisi karakter yang terorganisir. Kami kemudian menerapkan model U-Net untuk mendeteksi karakter-karakter potensial, seperti yang ditunjukkan pada Gambar 12d, dengan area berwarna merah yang mengindikasikan wilayah karakter potensial. Khususnya, dengan menggunakan Algoritma 2 (Detect-Other-Characters), model U-Net mengidentifikasi wilayah yang terlewatkan oleh YOLOv5, dan berhasil mendeteksi karakter Mandarin Sederhana dalam kasus ini (Gambar 12e).



Gambar 12. Campuran hasil deteksi skrip tulisan tangan.

Meskipun kemampuan untuk mendeteksi karakter tambahan mungkin tidak secara signifikan meningkatkan akurasi pendeteksian aksara Nusantara, hasilnya mengungkapkan wawasan yang menarik. Meskipun dilatih secara eksklusif untuk aksara Nusantara, model U-Net menunjukkan kemampuan untuk mengenali aksara lain, bahkan aksara yang tidak termasuk dalam kumpulan aksara khusus mereka. Hal ini serupa dengan model U-Net yang dilatih hanya untuk mendeteksi aksara Nusantara, namun hasilnya menunjukkan kemampuannya untuk mengidentifikasi aksara Latin, Arab, Korea, Jepang, dan Cina.

3.4. Diskusi

YOLOv5 adalah model pendeteksi karakter yang memiliki kemampuan yang kuat tetapi kesulitan dengan pengaturan aksara Nusantara yang tepat. YOLOv5 memiliki box loss sekitar 0.05, object loss mendekati 0.15, dan class loss mendekati 0.005, yang menunjukkan bahwa model ini tidak selalu dapat mendeteksi semua karakter. Model U-Net secara konsisten meningkatkan akurasi tetapi menunjukkan peningkatan yang stagnan dalam akurasi validasi dalam menghasilkan masker, yang menunjukkan adanya overfitting. Dalam percobaan, YOLOv5 mungkin melewati beberapa karakter aksara Nusantara,

sekitar 3% dari dokumen. Masalah ini mungkin muncul karena adanya gambar beresolusi rendah dan latar belakang yang rumit, yang sering dijumpai pada naskah-naskah lama. Dalam pengujian awal dengan

naskah Nusantara digital, model ini menunjukkan pendeteksian naskah yang sempurna. Namun, ketika noise diperkenalkan, kemampuan deteksi model menurun sebesar 10%. Bahkan model U-Net tidak menghasilkan peningkatan yang signifikan pada dataset yang berderau, yang mengindikasikan bahwa derau memang berdampak pada kinerja model U-Net. Model U-Net mengatasi beberapa kesalahan, dengan hanya meningkatkan 1,2% dari keseluruhan dokumen, dan terkadang mengidentifikasi karakter dari naskah lain. Hal ini menyoroti efisiensi pendeteksian lokasi karakter menggunakan YOLOv5 dan U-Net untuk naskah Nusantara.

4. KESIMPULAN

Dalam penelitian ini, kami mengembangkan model pendeteksi aksara Nusantara dengan menggunakan YOLOv5 dan U-Net. Namun, pengaturan karakter YOLOv5 tidak sesuai dengan konvensi penulisan aksara Nusantara yang benar. Kami memperkenalkan fungsi prediksi kalimat dengan menggunakan teknik pengelompokan dan algoritma pengaturan untuk pengaturan karakter yang sistematis untuk mengatasi hal ini. Meskipun YOLOv5 berkinerja baik, beberapa karakter tidak terdeteksi, sehingga mendorong kami untuk menggabungkan model U-Net, yang meningkatkan tingkat deteksi, termasuk karakter dari naskah yang tidak terlatih.

Hasil percobaan menunjukkan YOLOv5 mencapai kehilangan deteksi lokasi karakter sekitar 0,05 untuk kehilangan kotak, 0,15 untuk kehilangan objek, dan 0,005 untuk kehilangan kelas, sementara model U-Net menunjukkan akurasi mulai dari 75% hingga 90% dalam memprediksi wilayah karakter. Meskipun YOLOv5 tidak mencapai pendeteksian aksara Nusantara yang sempurna, mengintegrasikan model U-Net meningkatkan pendeteksian sebesar 1,2%. Meskipun penelitian ini telah mengembangkan deteksi aksara Nusantara, penelitian lebih lanjut diperlukan untuk menyelesaikan pembacaan aksara. Penelitian selanjutnya dapat mengeksplorasi penyempurnaan model pendeteksian, menambah jumlah dataset, termasuk pendeteksian arah karakter dan derajat kalimat, untuk meningkatkan hasil pembacaan.

REFERENSI

- [1] P. P. Charles, V. Harish, M. Swathi, dan C. H. Deepthi, "Sebuah tinjauan tentang berbagai teknik yang digunakan untuk pengenalan karakter optik," *International Journal of Engineering Research and Applications*, vol. 2, no. 1, hal. 659-662, 2012.
- [2] G. Nagy, S. Seth, dan M. Viswanathan, "Sebuah Prototipe Sistem Analisis Gambar Dokumen untuk Jurnal Teknis," *Computer*, vol. 25, no. 7, hal. 10-22, 1992.
- [3] D. R. Dickson dan K. Nusair, "Perspektif SDM: Perburuan global akan talenta di era digital," *Worldwide Hospitality and Tourism Themes*, vol. 2, no. 1, hal. 86-93, 2010. doi: 10.1108/17554211011012612.
- [4] J. Lo Bianco, "Pentingnya kebijakan bahasa dan multibahasa untuk keanekaragaman budaya," *International Social Science Journal*, vol. 61, no. 199, hal. 37-67, 2010. doi: 10.1111/j.1468-2451.2010.01747.x.
- [5] Y. Yamashita, K. Higuchi, Y. Yamada, dan Y. Haga, "Klasifikasi karakter Kanji yang dicetak dengan tangan dengan metode pencocokan segmen terstruktur," *Pattern Recognition Letters*, vol. 1, no. 5-6, pp. 475-479, 1983.
- [6] G. Lee, J. H. Lee, dan J. Yoo, "Multi-level post-processing untuk pengenalan karakter Korea menggunakan analisis morfologi dan evaluasi linguistik," *Pengenalan Pola*, vol. 30, no. 8, hal. 1347-1360, 1997.
- [7] C. L. Liu, F. Yin, DH Wang, dan QF Wang, "Pengenalan karakter bahasa Mandarin tulisan tangan secara online dan offline: perbandingan pada basis data baru," *Pengenalan Pola*, vol. 46, no. 1, hlm. 155-162, 2013.
- [8] M. Avadesh dan N. Goyal, "Pengenalan karakter optik untuk bahasa Sansekerta menggunakan jaringan saraf konvolusi," dalam *Lokakarya Internasional IAPR ke-13 tentang Sistem Analisis Dokumen (DAS) 2018*, hlm. 447-452, 2018. doi: 10.1109/DAS.2018.27.
- [9] A. Prasetyadi, J. Saputra, I. Ramadhanti, A. D. Sripamaji, dan R. R. Amalia, "Pendekatan Jaringan Syaraf Tiruan Depthwise Separable Convolutional Neural Network Berbasis DCT Minimalis untuk Aksara Tugut," *Jurnal Dinda: Ilmu Data, Teknologi Informasi, dan Analisis Data*, vol. 3, no. 2, hal. 59-64, 2023.
- [10] S. Alghyaline, "Sistem Pengenalan Karakter Optik Bahasa Arab Tercetak menggunakan Deep Learning," *Journal of Computer Science*, vol. 18, no. 11, pp. 1038-1050, 2022. doi: 10.3844/jcsp.2022.1038.1050.
- [11] B. Kataria dan H. B. Jethva, "Pengenalan Karakter Optik Berbasis CNN-Bidirectional LSTM pada Naskah Sansekerta: Tinjauan Literatur Sistematis yang Komprehensif," *Jurnal Internasional Penelitian Ilmiah dalam Ilmu Komputer, Teknik dan Teknologi Informasi*, hlm. 1362-1383, 2019. doi: 10.32628/cseit2064126.
- [12] A. W. Mahastama dan L. D. Krisnawati, "Pengenalan karakter optik untuk aksara Jawa cetak menggunakan segmentasi profil proyeksi dan pengklasifikasi centroid terdekat," di *2020 Asia Conference on Computers and Communications (ACCC)*, hlm. 52 - 56, 2020. doi: 10.1109/ACCC51160.2020.9347895.
- [13] M. H. Faishal, M. D. Sulistiyono, dan A. F. Ihsan, "Deteksi Huruf Aksara Jawa Menggunakan Faster R-CNN," *Jurnal Kecerdasan Buatan dan Penambangan Data*, vol. 6 no. 2, 243-251, 2023.
- [14] N. Suciati, N. P. Sutramiani, and D. Siahaan, "LONTAR_DET: Metode Deteksi Karakter Bahasa Bali yang Padat dan Bervariasi Tinggi pada Naskah Lontar," *IEEE Access*, vol. 10, pp. 14600-14609, 2022.
- [15] B. Gašparović, G. Mauša, J. Rukavina, dan J. Lerga, "Mengevaluasi YOLOv5, YOLOv6, YOLOv7, dan YOLOv8 di Lingkungan Bawah Air: Apakah Ada Peningkatan Nyata?," dalam *Konferensi Internasional ke-8 tentang Teknologi Cerdas dan Berkelanjutan (SpliTech) 2023*, hlm. 1-4, Juni 2023.
- [16] T. AN Dang dan DT Nguyen, "Ekstraksi informasi end-to-end dengan penyematan tingkat karakter dan U-Net atensi multi-tahap," *arXiv preprint arXiv:2106.00952*, 2021.

- [17] A. Prasetyadi, J. Saputra, I. Kresna, dan I. Ramadhanti, "Pendekatan Deep Learning untuk Pengenalan Karakter Optik Naskah Nusantara," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 17, no. 3, 2023.
- [18] D. Ghosh, T. Dube, dan A. Shivaprasad, "Pengenalan Naskah-sebuah tinjauan," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, hal. 2142-2161, 2010. doi: 10.1109/TPAMI.2010.30.
- [19] E. Alfian, "Penggunaan Unsur Aksara Nusantara Pada Huruf Modern," *Jurnal Komunikasi Visual*, vol. 7, no. 1, hal. 42-48, 2014.
- [20] P. T. Daniels, "Dasar-dasar Gramatologi," *Journal of the American Oriental Society*, vol. 119, no. 4, hal. 727-731, Oktober-Desember 1990. doi: 10.2307/602899.
- [21] J. Chen, M. Xie, Z. Xing, C. Chen, X. Xu, L. Zhu dan G. Li, "Deteksi objek untuk antarmuka pengguna grafis: Pembelajaran kuno atau pembelajaran mendalam atau kombinasi?", dalam *Prosiding Pertemuan Bersama ACM ke-28 tentang Konferensi Rekayasa Perangkat Lunak Eropa dan Simposium tentang Dasar-dasar Rekayasa Perangkat Lunak*, Nov. 2020, hlm. 1202-1214.
- [22] V. Lempitsky, P. Kohli, C. Rother, dan T. Sharp, "Segmentasi Gambar dengan Bounding Box Prior," dalam *International Conference on Computer Vision (ICCV)*, IEEE, hal. 277-284, 2009.
- [23] J. Son, M. Baek, M. Cho, dan B. Han, "Pelacakan multi-objek dengan jaringan saraf konvolusional quadruplet," dalam *Prosiding Konferensi IEEE tentang Visi Komputer dan Pengenalan Pola*, pp. 5620-5629, 2017.
- [24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, CY Fu, dan AC Berg, "Ssd: Single shot multibox detector," dalam *Computer Vision-ECCV 2016: Konferensi Eropa ke-14, Amsterdam, Belanda, 11-14 Oktober 2016, Prosiding, Bagian I*, hlm. 21-37, 2016.
- [25] P. Jiang, D. Ergu, F. Liu, Y. Cai, dan B. Ma, "Sebuah tinjauan perkembangan algoritma Yolo," *Procedia Computer Science*, vol. 199, pp. 1066-1073, 2022.
- [26] P. P. Bharati dan A. Pramanik, "Teknik pembelajaran mendalam-R-CNN untuk menutupi R-CNN: sebuah survei," dalam *Kecerdasan Komputasi dalam Pengenalan Pola: Prosiding CIPR 2019*, hal. 657-668, 2020.
- [27] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, X. Tao, J. Fang, I. Imyhy, L. Lorna, Y. Zeng, C. Wong, A. V. D. Montes, Z. Wang, C. Fati, J. Nadar, Tertawa, D. UnglvKitDe, V. Sonck, T. Tkianai, Y. YxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, dan M. Jain, "ultralytics/yolov5: v7.0-yolov5 sota realtime instance segmentation," Zenodo, 2022.
- [28] J. Redmon, S. Divvala, R. Girshick, dan A. Farhadi, "Anda hanya melihat sekali: Deteksi objek secara terpadu dan real-time," dalam *Prosiding konferensi IEEE tentang visi komputer dan pengenalan pola*, hal. 779-788, 2016.
- [29] C. Y. Wang, A. Bochkovskiy, dan H. Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," dalam *Prosiding Konferensi IEEE/CVF tentang Visi Komputer dan Pengenalan Pola*, 2021, hlm. 13029-13038.
- [30] A. Neubeck dan L. Van Gool, "Penekanan non-maksimum yang efisien," dalam *Konferensi Internasional ke-18 tentang Pengenalan Pola (ICPR'06)*, 2006, hal. 850-855.
- [31] O. Ronneberger, P. Fischer, dan T. Brox, "U-Net: Jaringan Konvolusional untuk Segmentasi Citra Biomedis," *arXiv: 1505.04597*, 2015.
- [32] A. M. Iktun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, dan J. Heming, "Algoritma pengelompokan K-means: Tinjauan komprehensif, analisis varian, dan kemajuan di era data besar," *Ilmu Informasi*, 2022.
- [33] R. H. Pramestya, "Deteksi dan Klasifikasi Kerusakan Jalan Aspal menggunakan Metode YOLO berbasis Citra Digital," 2018. [Online]. Available: <http://repository.its.ac.id/id/eprint/59044>.
- [34] I. van Kinsbergen, "Batu bertulis di Kawali dekat Tjiamis," dalam *Koleksi Gambar Digital KITLV, KITLV, Sebelum tahun 1900*. [Online]. Tersedia: <https://digitalcollections.universiteitleiden.nl/view/item/770870>.