Gun Detection Based On Deep Learning

Abstrak—Pada umumnya pendeteksian senjata adalah masalah yang sangat serius dan intensif dalam hal keamanan dan keselamatan publik, pendeteksian senjata adalah tugas yang sulit bahkan merepotkan ketika pengamat harus membuat secara otomatis atau dengan beberapa model kecerdasan buatan. Berbagai model pendeteksian objek yang berbeda sudah diimplementasikan dalam bidang instrumentasi. Tetapi dalam kaitannya dengan pendeteksian senjata, sulit untuk mendeteksi senjata dengan ukuran dan bentuk serta warna latar belakang yang berbeda. Saat ini, banyak pendekatan deep learning berbasis Convolutional Neural Network (CNN) yang dianjurkan untuk pengenalan dan klasifikasi secara real-time. Dalam jurnal ini, kami telah melakukan penelitian terhadap dua versi yang merupakan model canggih yang disebut YOLOV3 dan YOLOV4 untuk pendeteksian senjata. Untuk tujuan penelitian, kami merancang bangun alat pendeteksian senjata menggunakan raspeberry pi 4 serta membuat kumpulan data senjata dan gambar yang dikumpulkan dari gambar google serta sejumlah aset yang berbeda. Kami secara manual menganotasi gambar satu per satu dalam format yang berbeda karena YOLO membutuhkan file anotasi dalam format teks dan beberapa model lain memerlukan file anotasi dalam format XML. Kami melatih kedua versi tersebut pada kumpulan data senjata yang besar dan kemudian menguji hasilnya untuk analisis komparatif. Kami menjelaskan dalam artikel ini bahwa kinerja YOLOV4 secara signifikan mengungguli YOLOV3 dalam hal waktu pemrosesan dan sensitivitasnya, kami dapat membandingkan keduanya dalam hal metrik akurasi.

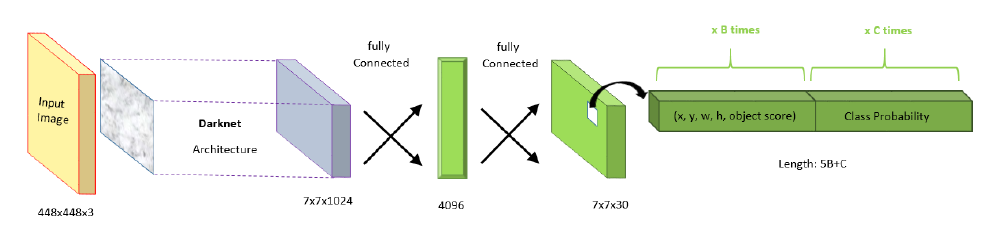
Ketentuan Indeks- Ketentuan Indeks—Deteksi senjata, Deteksi objek, Pengawasan visual, CNN keluarga YOLO

1. **PENDAHULUAN**

Kekerasan senjata adalah masalah hak asasi manusia dan kebebasan yang sangat serius di dunia. Hak asasi manusia yang utama ditakuti oleh kekerasan yaitu terkait kekerasan senjata. Menurut statistik, sekitar 500 orang meninggal setiap hari akibat kekerasan senjata. Lebih dari 44% pembunuhan di seluruh dunia terkait dengan kekerasan senjata. Antara 2012 dan 2016 lebih dari 1,4 juta orang tewas akibat kekerasan senjata [1], [28], [32]. Senjata biasanya digunakan untuk kekerasan dari pada membela diri [2]. Masalah ini membutuhkan alat dan teknik modern untuk menangani kosekuensi dari kekerasan senjata, seperti sebagian besar negara yang telah menggunakan sistem pengawasan video untuk memantau orang-orang di sekitar keramaian dari terorisme dan kejahatan [3]. Dengan menggunakan deep learning dan dalam specific convolutional neural network (CNN), kita dapat mengestimasi serta mengevaluasi objek dalam citra melalui klasifikasi dan lokalisasi yang disebut object detection [4]. Ada banyak pemanfaatan dan penerapan deteksi objek, [34], [26], [27], [28] seperti deteksi wajah [5], [27], [29] deteksi pejalan kaki [6], [30], [ 31] deteksi kerangka [7]. Ada banyak arsitektur dan algoritma yang hadir seperti YOLO dan versinya [8], [9], [10], [11]. R-CNN dan versinya [12], [13], [14], [15]. Girshick et al memperkenalkan CNN yang berbasis wilayah disebut R-CNN, di mana CNN digabungkan dengan algoritma proposal wilayah. Pendekatan pencarian selektif digunakan untuk mengekstraksi 2000 wilayah, kemudian klasifikasi akan berlaku hanya pada wilayah yang diekstraksi dari seluruh citra. Setelah itu, ada beberapa perbaikan yang dilakukan oleh peneliti untuk mengatasi keterbatasan algoritma yang ada dengan memberikan seluruh gambar sebagai input ke jaringan, bukan proposal wilayah, dan kemudian peta fitur konvolusional memberikan identifikasi proposal wilayah. Kemudian Faster R-CNN diusulkan oleh Shaoqing Ren et al. dengan mengubah teknik pencarian selektif menjadi algoritma deteksi objek [33]. Dalam makalah ini, kami merancang alat pendeteksian senjata dan membandingkan kinerja berbagai model. Kami memeriksa pendekatan saat ini dari dua versi yang menggunakan YOLOV4 dan YOLOV3 dengan menggunakan mikrokontroller raspberry pi 4 dengan gabungan kamera Razer Kiyo. Kami juga telah menghasilkan kumpulan data baru yang terdiri dari 7800 gambar senjata yang juga merupakan bagian dari penelitian dan eksplorasi ini. Semua gambar dikumpulkan dari internet dan aset utama adalah gambar google, video CCTV, dan film. Setelah berhasil mengumpulkan gambar-gambar ini, kami memberi label gambar-gambar ini secara manual dengan bantuan alat LabelImg[33]. Kebenaran dasar dari setiap senjata tersedia bersama dengan gambar dalam kumpulan data yang diberikan. Dataset yang dibuat juga berisi berbagai format seperti YOLO, Faster R-CNN, Pascal VOC bersama dengan kotak pembatas yang diputar.

Manfaat penting dari studi penelitian ini diberikan di bawah ini.

* menyajikan perbandingan algoritma canggih YOLO dalam hal deteksi senjata. Kami juga membahas sensitivitas objek dalam dua jenis kelas senjata yang berbeda yaitu pistol dan pistol.
* Kami juga mengembangkan kumpulan data senjata baru yang terdiri dari 7800 gambar dan sebagian besar gambar berisi lebih dari satu senjata. Kotak pembatas berorientasi sudut juga tersedia dalam kumpulan data ini.
* Dataset disediakan dalam berbagai format seperti YOLO, Faster-RCNN, Pascal-VOC. Ini akan membantu para peneliti untuk melakukan penelitian yang bermanfaat di masa depan.



Gambar 1: Arsitektur Jaringan YOLO untuk Mewakili Implementasi Convolutional Layers

Pada bagian ini, kami mengukur kinerja serta mengamati pendekatan versi YOLOV4 dan YOLOV3 dalam hal pendeteksian senjata. bab selanjutnya disusun sebagai berikut: Bagian-II menjelaskan pekerjaan terkait tentang deteksi senjata. Bagian-III menyampaikan perbandingan antara YOLOV4 dan YOLOV3 untuk pendeteksian senjata. Bagian-IV menyimpulkan penelitian dan mendiskusikan hasilnya.

1. **PEKERJAAN YANG BERHUBUNGAN**

Bagian ini berisi informasi tentang topik pendeteksian senjata yang dibahas oleh Convolutional Neural Networks.

Elmir et al.[16] memperkenalkan model yang bekerja dalam langkah-langkah yang berbeda. Langkah pertama adalah akuisisi citra atau pengambilan gambar, yang kedua adalah deteksi gerakan, dan yang terakhir adalah deteksi senjata. Mereka melakukan percobaan pada tiga model yang berbeda, yang pertama adalah model berbasis CNN, yang kedua adalah model berbasis Fast R-CNN, dan yang ketiga adalah model berbasis CNN Mobile-Net. Untuk evaluasi ketiga model ini, jumlah database yang digunakan untuk tahap pembelajaran adalah 2. Untuk tugas klasifikasi, data set pelatihan berisi 9261 gambar dengan 102 kelas dimana kelas pistol di 200. Untuk tugas wilayah, data set pelatihan berisi 3000 gambar. Set data pengujian untuk deteksi dan klasifikasi berisi 608 citra dengan 304 adalah pistol. Mereka menggunakan sampel sebanyak 420 gambar model pembelajaran dari database untuk teknik usulan wilayah. Mereka menguji model pertama dengan 608 gambar, model kedua dengan 200 gambar, dan model ketiga dengan 420 gambar. Jika kita katakan tentang hasilnya, mereka mendapat akurasi 55% pada model pertama, 80% pada model kedua, dan 90% pada model ketiga.

Dalam penelitian ini, metode Convolutional Neural Networks (RCNN) berbasis Region dan Fast Region berdasarkan Convolutional Neural Networks (FRCNN) diimplementasikan untuk deteksi otomatis senjata dalam video pengawasan [17]. Algoritma pendeteksian objek juga digunakan untuk pendeteksian pisau dalam video yang diberikan[18].

***A. Penjelasan YOLO***

YOLOV4[11]. adalah hasil perbaikan dari versi sebelumnya diantaranya YOLOV1[8], YOLOV2[9] dan YOLOV3[10].

1. ***Algoritma YOLOV1***

YOLO terdiri dari 24 convolutional layer dengan 2 fully connected layer. Beberapa lapisan menggunakan konvolusi ukuran 11 untuk mengurangi kedalaman peta fitur. Varian lain menggunakan 9 lapisan konvolusional yang disebut Fast YOLO yang paling memengaruhi akurasi[23].

YOLO membagi gambar yang diberikan ke dalam kisi . Sel kisi dapat berkorelasi dengan satu objek dan memprediksi kotak pembatas yang jumlahnya tetap. Setiap kotak juga diberi nilai kepercayaan sehingga berikut adalah detail yang juga diprediksi untuk semua kotak pembatas (X, Y, W, H, Keyakinan Skor). Untuk estimasi klasifikasi objek, sel grid berkaitan dengan jumlah probabilitas kelas dari kelas model yang dilambangkan dengan . Tema utama di balik YOLOV1 adalah membuat jaringan CNN tunggal untuk prediksi.[24]

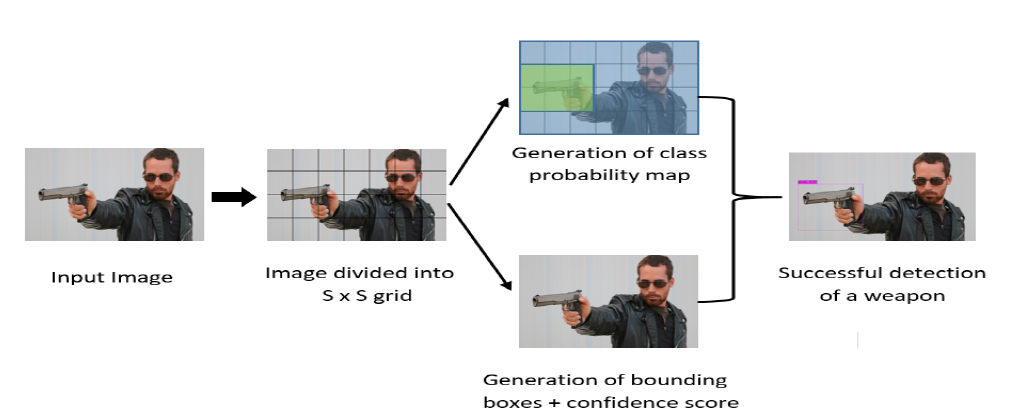
* menunjukkan jumlah sel kisi yang dikandung oleh yang diberikan gambar.
* menunjukkan kotak pembatas yang terdapat di setiap sel kisi.
* menunjukkan jumlah kelas untuk pelatihan.

Ketika Anda memiliki informasi tentang prediksi dan pengkodeannya, bagian lainnya mudah dipahami. Struktur YOLO mirip dengan CNN, yang berisi lapisan convolutional dan lapisan max-pooling bersama dengan 2 lapisan yang terhubung sepenuhnya pada akhirnya.[25]

1. ***Versi YOLOV2 yang ditingkatkan.***

Penantang lain untuk YOLO diperkenalkan, dinamakan sebagai SSD[20] (Single Shot MultiBox Detector). SSD mengalahkan YOLO untuk deteksi objek real-time dalam hal akurasi. Oleh karena itu, versi YOLO yang ditingkatkan diperkenalkan dengan banyak perbaikan untuk meningkatkan waktu dan akurasi pemrosesan.

Peningkatan pertama YOLOV2 dibandingkan YOLOV1 adalah teknik Batch Normalization (BN)[21], yang disajikan pada tahun 2015. Teknik ini digunakan untuk penskalaan dan penyesuaian aktivasi untuk menormalkan lapisan masukan. Terlihat bahwa mAP meningkat sebesar 2% saat menerapkan BN ke semua lapisan konvolusional di YOLO.

Gambar 2: Arsitektur YOLO untuk Mewakili Tindakan yang Dilakukan pada Gambar

Peningkatan kedua dalam versi ini adalah menerapkan Pengklasifikasi Resolusi Tinggi. Melatih model pada gambar 224x224, untuk menyempurnakan versi ini juga menerapkan gambar 448x448 di jaringan klasifikasi. Jadi mAP meningkat sebesar 4%. Peningkatan ketiga pada versi ini adalah implementasi konvolusi dengan kotak jangkar. Versi ini menghapus seluruh Lapisan yang terhubung dan menggunakan kotak jangkar untuk prediksi kotak pembatas. Untuk meningkatkan resolusi keluaran, satu lapisan penyatuan dikecualikan. Implementasi kotak jangkar mengurangi peta sebesar 0,3%, dengan kotak jangkar, penarikan kembali meningkat sebesar 7%.

Sedangkan perbaikan selanjutnya pada versi ini adalah dengan menerapkan cluster dimensi. Pengelompokan K-means digunakan untuk menemukan kotak jangkar yang benar. Untuk pengelompokan mereka menggunakan skor IOU bukan jarak Euclidean karena kotak yang lebih kecil menghasilkan lebih sedikit kesalahan daripada kotak yang lebih besar. Peningkatan kelima dalam versi ini adalah prediksi lokasi secara real-time. Versi sebelumnya tidak memiliki batasan prediksi lokasi yang mengakibatkan model menjadi tidak seimbang pada iterasi primer.

1. ***Versi yang ditingkatkan YOLOV3.***

Peningkatan pertama dari versi ini adalah penerapan klasifikasi multi-label. Untuk mengetahui probabilitas dari skor fungsi soft-max digunakan versi sebelumnya. Kerugian lintas-entropi biner digunakan untuk kerugian klasifikasi daripada menggunakan mean square error seperti pada versi sebelumnya. Peningkatan kedua dari versi ini adalah prediksi kotak pembatas yang berbeda. Versi ini memberikan satu kotak jangkar untuk setiap objek kebenaran dasar. Itu mengabaikan kotak pembatas tumpang tindih lainnya yang terjadi pada hasil ambang batas yang ditentukan.Peningkatan ketiga dari versi ini adalah prediksi lintas skala dengan bantuan fitur jaringan piramida. Versi ini menggunakan 3 skala berbeda untuk prediksi kotak dan kemudian mengekstrak fitur darinya.Peningkatan keempat dari versi ini adalah ekstraktor fitur yang disebut Darknet-53. Jumlah lapisan CNN adalah 53 yang mengikuti jaringan koneksi lewati. Ini juga menerapkan lapisan konvolusional 3x3 dan 1x1 dan meningkatkan akurasi dengan operasi floating-point yang lebih kecil.

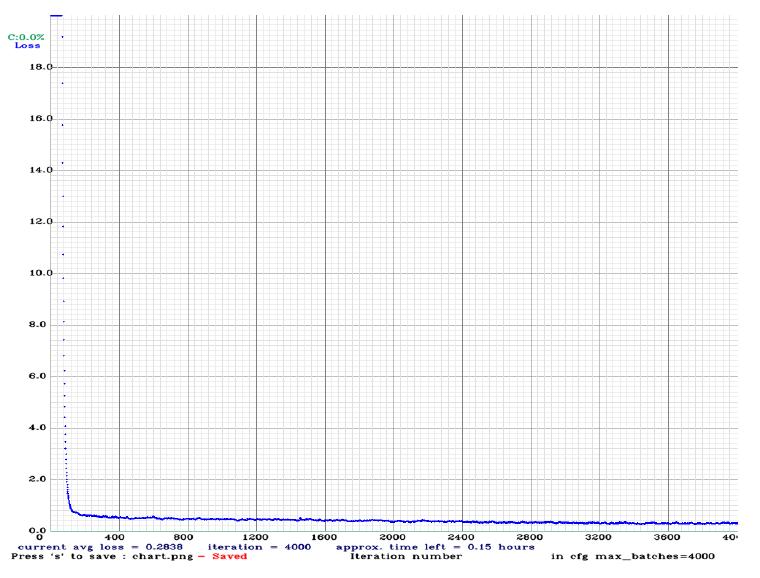
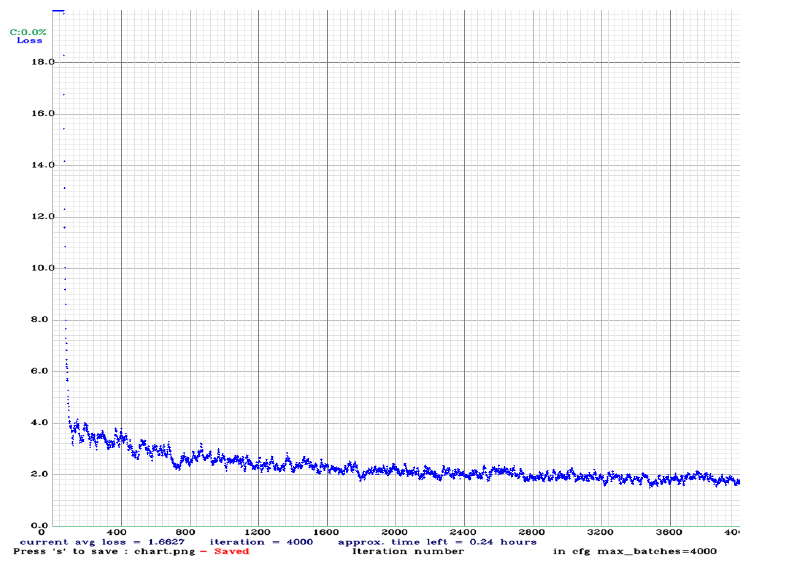
Ada 4 prediksi koordinat oleh jaringan untuk setiap kotak pembatas, tx, ty, tw, th. Jika sudut kiri atas gambar dilambangkan dengan ), tinggi dan lebar kotak pembatas dilambangkan dengan (Pw,Ph), maka prediksinya dinyatakan sebagai:

1. ***Versi yang ditingkatkan YOLOV4.***

Versi ini menggunakan koneksi parsial lintas tahap (CSP), tulang punggung terbaru untuk CNN untuk meningkatkan kemampuan belajar. Cross mini-Batch Normalization (CmBN) digunakan untuk membagi batch menjadi mini-batch. Self Adversarial Training (SAT) menghadirkan pendekatan augmentasi data lain yang bekerja di kedua tahap mundur ke depan. Fungsi aktivasi saraf non-monoton digunakan yang diatur sendiri yang dikenal sebagai Aktivasi mish. Untuk augmentasi, digunakan teknik yang menggabungkan 4 gambar pelatihan daripada satu gambar yang disebut augmentasi data mosaik. Teknik regularisasi yang lebih baik digunakan untuk CNN yang disebut regularisasi drop block. Untuk mencapai akurasi dan kecepatan untuk masalah regresi kotak pembatas, kerugian CIoU digunakan.

1. **PERBANDINGAN PENEMUAN FAKTA ANTARA YOLOV4 DAN YOLOV3**

Pada bagian ini, kita akan membahas tentang data pelatihan dan data pengujian dan juga mempertimbangkan perangkat keras dan perangkat lunak yang digunakan untuk eksperimen. Evaluasi model disediakan untuk pemahaman yang lebih baik tentang perbandingan. Hasil deteksi YOLOV4 dan YOLOV3 masing-masing ditunjukkan pada Gambar 4.



Gambar 3: Grafik Akurasi Validasi untuk YOLOV4 dan YOLOV3

TABEL I: STATISTIK TENTANG DATASET SENJATA

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Total Images | Gun | Total instances |
| Training | 6240 | 4384 | 7347 |
| Testing | 1561 | 1128 | 1904 |

***A. Penjelasan Dataset***

Untuk melakukan percobaan perbandingan, kami membuat dataset senjata yang dibagi menjadi set pelatihan dan set pengujian. Ada 6240 gambar dalam set pelatihan dan 7347 contoh senjata berlabel di mana 4384 contoh milik kelas senjata dan 2963 milik kelas pistol. Set pengujian berisi total 1561 gambar dan 1904 contoh senjata berlabel di mana 1128 contoh milik kelas senjata dan 776 milik kelas pistol. Kumpulan data ini dibuat setelah kumpulan gambar google dan banyak sumber lain seperti film dan gambar CCTV. Kami mengumpulkan gambar senjata dengan berbagai warna, latar belakang, ukuran, dan bentuk untuk memeriksa model secara inklusif. Setelah melakukan proses pengumpulan gambar yang melelahkan, kami membubuhi keterangan setiap gambar secara manual dengan bantuan alat anotasi yang dikenal sebagai LabelImg[35]. File anotasi tersedia dalam berbagai format.

Untuk YOLO kita membutuhkan file teks yang berisi parameter (x, y, w, h, class) "x" dan "y" menunjukkan titik pusat, "w" menunjukkan lebar kotak sedangkan "h" menunjukkan tinggi kotak dan "kelas" menyatakan bahwa objek milik kelas mana.

Format anotasi lainnya adalah Pascal-VOC dalam bentuk file XML yang berisi empat titik (xmin,ymin,xmax,ymax). Dataset kami juga tersedia dalam bentuk kotak pembatas berorientasi yang akan sangat membantu untuk studi deteksi senjata sadar orientasi. Ada dua jenis kelas yang termasuk dalam dataset kami, kelas pertama dikenal sebagai "Gun" dan yang kedua adalah "Pistol". Pembagian dataset menjadi data training dan data testing dengan perbandingan 80:20.

TABEL II: PERBANDINGAN KINERJA YOLO DENGAN DATASET YANG BERBEDA

|  |  |  |
| --- | --- | --- |
| Detector | mAP | Used Data-set |
| YOLOV3 | 77.30 | Weapons Data-set |

***B. Penjelasan Perangkat Lunak dan Perangkat Keras***

Sekarang kita bahas konfigurasinya, di YOLOV4 kita menggunakan konfigurasi default, pengoptimal yang digunakan dengan momentum 0.9, tinggi dan lebar keduanya diatur ke 512. Tingkat pembelajaran diatur ke 0.001, dan peluruhan diatur ke 0.0005. Ukuran batch diatur ke 64, jumlah max batch sama dengan 4000. Ambang batas untuk tumpang tindih kotak jangkar dengan kebenaran dasar diatur menjadi 0,7 dalam hal ini, jika nilai IOU lebih besar dari 0,7 maka kotak dipilih jika tidak maka akan diabaikan. Di YOLOV3 kami juga menggunakan momentum konfigurasi default dan kecepatan pembelajaran juga sama seperti di YOLOV4, tetapi tinggi dan lebarnya diatur ke 416 dan semua konfigurasi lainnya sama seperti di YOLOV4. Konfigurasi komputer diberikan di bawah ini:

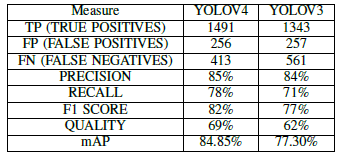
* Prosesor: Intel Core i7-9700
* Kartu Grafis: Nvidia Ge Force 1080 TI
* RAM: RAM 62 GB
* Sistem Operasi: Windows 10

***C. Evaluasi Kinerja***

Untuk analisis komparatif dari kedua versi, kami menggunakan parameter ini (PRECISION, RECALL, SKOR F1, KUALITAS, mAP) dan deskripsi parameter ini diberikan di bawah ini:

di mana TP (POSITIF BENAR) menunjukkan jumlah senjata yang terdeteksi oleh model. FP (FALSE POSITIVES) menunjukkan jumlah objek non-senjata yang terdeteksi secara salah sebagai senjata. FN (NEGATIF PALSU) menunjukkan jumlah senjata yang modelnya tidak dapat mengenalinya sebagai senjata.

TABEL III: Metrik Evaluasi YOLOV4 dan YOLOV3 pada Data Uji



***D. Perbandingan antara YOLOV4 dan YOLOV3***

Di bagian ini, kami akan membandingkan kedua versi berdasarkan metrik yang diberikan. Tabel II menunjukkan bahwa kedua versi memiliki nilai presisi yang tinggi dan nilai presisi yang tinggi ini menentukan kapan model mendeteksi objek sebagai senjata.

maka kemungkinan besar benda tersebut adalah senjata. Jadi, kapasitas model untuk mendeteksi senjata sebenarnya sangat tinggi. Kemungkinan untuk mendeteksi objek non senjata sebagai senjata sangat kecil. Anda dapat melihat perbedaan yang jelas dalam metrik evaluasi. Evaluasi ini menunjukkan bahwa kedua versi memiliki tingkat presisi yang tinggi (85% untuk V4 dan 84% untuk V3). Namun saat memeriksa penarikan, kami menemukan bahwa penarikan YOLOV4 lebih besar daripada YOLOV3 (78% untuk V4 vs 71% untuk V3). Penarikan kembali menunjukkan kapasitas model untuk mendeteksi semua contoh senjata dalam gambar yang diberikan. Jika nilai False Negatives tinggi, kemungkinan model melewatkan beberapa instance. Skor F1 memberikan gambaran umum tentang kekokohan model sehingga ada juga perbedaan antara Skor F1 dari dua versi. Anda juga dapat melihat perbedaan besar antara mAP dari kedua versi tersebut.

1. **KESIMPULAN**

Dalam makalah penelitian ini, analisis Komparatif telah dibuat untuk dua versi algoritma deteksi objek canggih yang dikenal sebagai YOLOV4 dan YOLOV3. Kami telah melakukan analisis komparatif pencarian fakta untuk tugas deteksi senjata. Kami mengambil awal dari garis besar kedua versi tersebut, melihat arsitektur dan perbaikan dari versi sebelumnya. Sejak saat itu, kami membuat analisis komparatif dengan bantuan kumpulan data senjata buatan sendiri yang independen. Dataset dibagi menjadi set pelatihan dan set pengujian, baik versi yang dilatih pada set data tersebut dan selanjutnya mengukur kinerja pada set data yang diberikan. Performa diperkirakan berdasarkan parameter yang diberikan, misalnya Presisi, Recall, Skor F1, Kualitas, mAP, dan sebagainya. Kami telah menunjukkan bahwa kinerja YOLOV4 jelas lebih unggul dari YOLOV3 dan menyoroti hal-hal di balik peningkatan tersebut. Perbandingan ini memberi para peneliti pengaturan yang luar biasa untuk melihat sesuatu secara mendalam dan memberikan informasi bahwa perubahan kecil memberikan hasil yang lebih baik. Untuk pekerjaan di masa mendatang, kami akan membuat pengukuran untuk meningkatkan gambar dalam kumpulan data kami dan selanjutnya meningkatkan ukuran kelas untuk memperluas deteksi senjata.

[1] https://www.amnesty.org/en/what-we-do/arms-control/gun-violence/

[2] https://www.hsph.harvard.edu/hicrc/firearms-research/gun-threats-andself-

defense-gun-use-2/

[3] https://www.ifsecglobal.com/video-surveillance/role-cctv-cameraspublic-

privacy-protection/

[4] Z. Zhao, P. Zheng, S. Xu and X. Wu, ”Object Detection With Deep

Learning: A Review,” in IEEE Transactions on Neural Networks and

Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi:

10.1109/TNNLS.2018.2876865.

[5] Rein-Lien Hsu, M. Abdel-Mottaleb and A. K. Jain, ”Face detection

in color images,” in IEEE Transactions on Pattern Analysis and

Machine Intelligence, vol. 24, no. 5, pp. 696-706, May 2002, doi:

10.1109/34.1000242.

[6] M. Szarvas, A. Yoshizawa, M. Yamamoto and J. Ogata, ”Pedestrian detection

with convolutional neural networks,” IEEE Proceedings. Intelligent

Vehicles Symposium, 2005., Las Vegas, NV, USA, 2005, pp. 224-229,

doi: 10.1109/IVS.2005.1505106.

[7] X. Bai, Xinggang Wang, L. J. Latecki, W. Liu and Z. Tu, ”Active

skeleton for non-rigid object detection,” 2009 IEEE 12th International

Conference on Computer Vision, Kyoto, 2009, pp. 575-582, doi:

10.1109/ICCV.2009.5459188.

[8] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only

look once: Unified, real-time object detection,” in 2016 IEEE Conference

on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas,

NV, USA, June 27-30, 2016, pp. 779–788, 2016.

[9] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in 2017

IEEE Conference on Computer Vision and Pattern Recognition, CVPR

2017, Honolulu, HI, USA, July 21-26, 2017, pp. 6517–6525, 2017.

[10] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,”

CoRR, vol. abs/1804.02767, 2018.

[11] Alexey Bochkovskiy et al., YOLOv4: Optimal Speed and Accuracy of

Object Detection, Apr 2020.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature

hierarchies for accurate object detection and semantic segmentation,” in

Proceedings of the IEEE Computer Society Conference on Computer

Vision and Pattern Recognition, pp. 580–587, 2014.

[13] R. Girshick, “Fast R-CNN,” in Proceedings of the IEEE International

Conference on Computer Vision, 2015.

[14] R. Girshick, “Fast R-CNN,” in Proceedings of the IEEE International

Conference on Computer Vision, 2015.

[15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards RealTime

Object Detection with,” IEEE TRANSACTIONS ON PATTERN

ANALYSIS AND MACHINE INTELLIGENCE, 2017.

[16] Elmir, Youssef, Sid Ahmed Laouar, and Larbi Hamdaoui. ”Deep Learning

for Automatic Detection of Handguns in Video Sequences.” JERI.

2019.

[17] Olmos R., Tabik S., Herrera F., “Automatic handgun detection alarm

in videos using deep learning,” Neurocomputing, vol. 275, pp. 66-72,

February 2018.

[18] Buckchash H., et al., “A robust object detector: application to detection

of visual knives,” 2017 IEEE International Conference on Multimedia

Expo Workshops (ICMEW), July 2017.

[19] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for

Large-Scale Image Recognition,” International Conference on Learning

Representations (ICRL), 2015.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C.

Berg, “SSD: Single shot multibox detector,” in Lecture Notes in Computer

Science (including subseries Lecture Notes in ArtificialIntelligence and

Lecture Notes in Bioinformatics), 2016.

[21] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep

network training by reducing internal covariate shift,” CoRR, vol.

abs/1502.03167, 2015.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for

Image Recognition,” Arxiv.Org, 2015.

[23] M. Haroon, M. Shahzad, M.M. Fraz , ”Multi-sized Object Detection

Using Spaceborne Optical Imagery”, IEEE Journal of Selected Topics in

Applied Earth Observations and Remote Sensing (JSTARS), Vol. 0 , No.

0, PP. , Jun, 2020.

[24] I Khurram, M.M. Fraz , M Shahzad, NM Rajpoot , ”Dense-CaptionNet:

A Sentence Generation Architecture for Fine-Grained Description of

Image Semantics”, Cognitive Computing, Vol. 12 , No. 2, PP. 1-31, Mar,

2020.

[25] S. B. Ahmed, S. F. Ali, J. Ahmad, M. Adnan, M. M. Fraz , ”On

the Frontiers of Pose Invariant Face Recognition: A Review”, Artificial

Intelligence Review, Vol. 2019 , No. 1, PP. 1-64, Jul, 2019.

[26] R M S Bashir, M Shahzad, M M Fraz , ”VR-PROUD: Vehicle Reidentification

using PROgressive Unsupervised Deep architecture”, Pattern

Recognition, Vol. 90 , No. 1, PP. 52-65, Jan, 2019.

[27] N. Pervaiz, M. M. Fraz, M. Shahzad , ”Person Re-Identification Using

Hybrid Representation Reinforced by Metric Learning”, IEEE Access,

Vol. 7 , No. 1, Dec, 2018

[28] N. Perwaiz, M.M. Fraz, M. Shahzad , ”Smart Visual Surveillance:

Proactive Person Re-identification instead of Impulsive Person Search”,

Proceedings of the 23rd IEEE International Multitopic Conference (INMIC

2020), Nov, 2020, Bahawalpur

[29] N. Pervaiz, M. M. Fraz, M. Shahzad , ”Hierarchical Refined Local

Associations for Robust Person Re-Identification”, Proceedings of the International

Conference on Robotics and Automation in Industry (ICRAI),

Oct, 2019, Islamabad , Pakistan.

[30] S. Batool, M. Z. Ali ; M. Shahzad and M. M. Fraz , ”End to End Person

Re-Identification for Automated Visual Surveillance”, Proceedings of the

International Conference on Image Processing, Applications and Systems

(IPAS), Dec, 2018, Sophia Antipolis , France.

[31] W. Anser, M M Fraz, M Shahzad , , ”Two Stream Deep CNN-RNN

Attentive Pooling Architecture for Video-based Person Re-identification”,

Proceedings of the 23rd Iberoamerican Congress on Pattern Recognition,

Nov, 2018, Madrid , Spain.

[32] R M S Bashir, M Shahzad, M M Fraz , , ”DUPL-VR: Deep Unsupervised

Progressive Learning for Vehicle Re-Identification”, Proceedings of the

13th International Symposium on Visual Computing, Nov, 2018, Las

Vegas , United States.

[33] I Khurram, M M Fraz, M Shahzad , , ”Detailed Sentence Generation

Architecture for Image Semantics Description”, Proceedings of the 13th

International Symposium on Visual Computing, Nov, 2018, Las Vegas ,

United States.

[34] SKJ Rizwi, MA Azad, M.M. Fraz , ”Spectrum of Advancements and

Developments in Multidisciplinary Domains for Generative Adversarial

Networks (GANs)”, Archives of Computational Methods in Engineering,

Vol. 2021 , No. 1, Apr, 2021

[35] “LabelImg,”https://github.com/tzutalin/labelImg.

**Lampiran code:**

import cv2

import numpy as np

import os

import time

import pyautogui

import imutils

from datetime import datetime

file\_name = 'hasil\_data.csv'

my\_file = open(file\_name, 'w', encoding='utf-8')

#f = open("test.txt", "w")

net = cv2.dnn.readNet('yolov3\_training\_final.weights', 'yolov3-tiny.cfg')

net.setPreferableBackend(cv2.dnn.DNN\_BACKEND\_CUDA)

net.setPreferableTarget(cv2.dnn.DNN\_TARGET\_CUDA)

classes = []

with open("classes.txt", "r") as f:

classes = f.read().splitlines()

cap = cv2.VideoCapture('vid\_test/gun\_test1.mp4')

font = cv2.FONT\_HERSHEY\_PLAIN

colors = np.random.uniform(0, 255, size=(100, 3))

frame\_width = int(cap.get(3))

frame\_height = int(cap.get(4))

size = (frame\_width, frame\_height)

result = cv2.VideoWriter('result.avi',

cv2.VideoWriter\_fourcc(\*'MJPG'),

10, size)

while True:

\_, img = cap.read()

height, width, \_ = img.shape

blob = cv2.dnn.blobFromImage(img, 1/255, (416, 416), (0,0,0), swapRB=True, crop=False)

net.setInput(blob)

output\_layers\_names = net.getUnconnectedOutLayersNames()

layerOutputs = net.forward(output\_layers\_names)

boxes = []

confidences = []

class\_ids = []

for output in layerOutputs:

for detection in output:

scores = detection[5:]

class\_id = np.argmax(scores)

confidence = scores[class\_id]

if confidence > 0.2:

center\_x = int(detection[0]\*width)

center\_y = int(detection[1]\*height)

w = int(detection[2]\*width)

h = int(detection[3]\*height)

x = int(center\_x - w/2)

y = int(center\_y - h/2)

boxes.append([x, y, w, h])

confidences.append((float(confidence)))

class\_ids.append(class\_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.2, 0.4)

daftar = []

if len(indexes)>0:

for i in indexes.flatten():

x, y, w, h = boxes[i]

confidence\_round = round(confidences[i],2)\*100

confidence\_print = ("%.2f" % confidence\_round)

label = str(classes[class\_ids[i]])

daftar.append(label)

data = str(daftar)

center\_rect = (center\_x,center\_y)

#confidence = str(round(confidences[i],2)\*100)

color = colors[i]

cv2.rectangle(img, (x,y), (x+w, y+h), color, 2)

cv2.putText(img,label, (x,y), font, 1, (0,255,0), 2)

cv2.putText(img,confidence\_print, (x+50,y), font, 1, (0,255,0), 2)

cv2.circle(img, center\_rect, radius=1, color=(0, 0, 255), thickness=2)

#with open(filename, 'w', encoding='utf-8') as my\_file:

my\_file.write(label+","+confidence\_print+","+str(center\_x)+","+str(+center\_y) +'\n')

#print(my\_file.closed) # 👉️ False

result.write(img)

print(daftar)

cv2.imshow('Image', img)

key = cv2.waitKey(1)

if key==27:

break

my\_file.close()

result.release()

cap.release()

cv2.destroyAllWindows()