Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 2, April 2025, Page. 951-964
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4224

You Only Look Once v5 and Long Short-Term Memory Implementation for Crowd Anomaly Detection

Nicholas Chrisandy[1], Ken Ratri Retno Wardani*[2], Inge Martina[3], and Hery Heryanto[4]

[1,2,3,4]Informatics, Institut Teknologi Harapan Bangsa, Indonesia

Email: ken_ratri@ithb.ac.id

Received : Dec15, 2024; Revised : Mar 18, 2025; Accepted : Apr 25, 2025; Published : May 17, 2025

Abstract

In Indonesia, 116,000 traffic accidents and 370,747 workplace accidents occurred in 2023, emphasizing the urgent need for effective surveillance systems for monitoring crowded areas such as public sidewalks, roads, workplaces, and school hallways. This study introduces a novel approach combining You Only Look Once v5 (YOLOv5) and Long Short-Term Memory (LSTM) networks for crowd anomaly detection. Unlike traditional methods, this hybrid framework utilizes YOLOv5 for precise feature extraction from video frames and LSTM to capture temporal dependencies for detecting anomalous behaviors. The dataset used includes scenes from the Crowd Anomaly Detection UML Dataset, consisting of a 1-minute and 11-second video extracted into 852 images. Hyperparameter tuning was conducted for epochs and learning rates in the YOLOv5 model, as well as for epochs and units in the LSTM model. The proposed framework achieved remarkable results, with 98% accuracy, 100% precision, and 86% F1-Score. However, improvements in class distribution within the training data could enhance model performance further. These findings demonstrate the potential of the proposed method for real-world applications in improving public safety and effective anomaly detection. This research proves that the proposed method which uses separate feature extraction method before detecting anomaly provides a better result in crowd anomaly detection.

Keywords : *Crowd Anomaly Detection, Long Short-Term Memory, Surveillance systems, You Only Look Once v5.*

1. PENDAHULUAN

Dalam dua puluh tahun ke depan, aplikasi pengawasan berbasis kamera akan menjadi salah satu tantangan bagi pengelolaan kota pintar dalam upaya meningkatkan kualitas hidup masyarakat. Ares dan Kulshrestha memanfaatkan perangkat seluler [1] dan perangkat pintar [2] untuk mendeteksi kerumunan dan menganalisis pergerakannya, tetapi penelitian tersebut belum mendeteksi anomali.

Pada tahun 2023 di Indonesia terjadi 116.000 kasus kecelakaan lalu lintas [3] dan terjadi 370.747 kasus kecelakaan kerja [4]. Kejadian lainnya juga seperti tindak kejahatan dan tawuran merupakan hasil dari perilaku anomali [5], [6]. Peristiwa seperti ini seharusnya dapat dicegah dengan adanya CCTV yang dapat menginformasikan jika terjadi anomali pada kerumunan dan memberikan peringatan dini [7], [8]. Selama ini CCTV hanya dimanfaatkan untuk merekam kejadian, mengenali objek, dan gerakan. Berdasarkan data yang telah disampaikan, diperlukan sebuah sistem cerdas yang mampu mendeteksi anomali tersebut [9], [10], [11].

Untuk mengatur dan mengamankan pejalan kaki, pelacakan masing-masing orang sangat penting untuk memberikan gambaran mengenai keadaan dari kerumunan [12], [13], [14]. Dengan memonitor kerumunan di tempat umum dapat membantu pihak keamanan dengan memberikan peringatan jika terdapat anomali, hal ini dapat mencegah korban cedera dan korban jiwa [15][16].

Li, 2020 mengusulkan sebuah model yang menggunakan strategi *multi-stage clustering* untuk mengumpulkan informasi pergerakan secara akurat. Li menyatakan bahwa modelnya masih perlu diuji

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4224

dalam sebuah aplikasi dari kamera secara *real-time*. Tingkat akurasi model sudah mencapai 87%, namun ada kendala yaitu biaya komputasi yang besar [17].

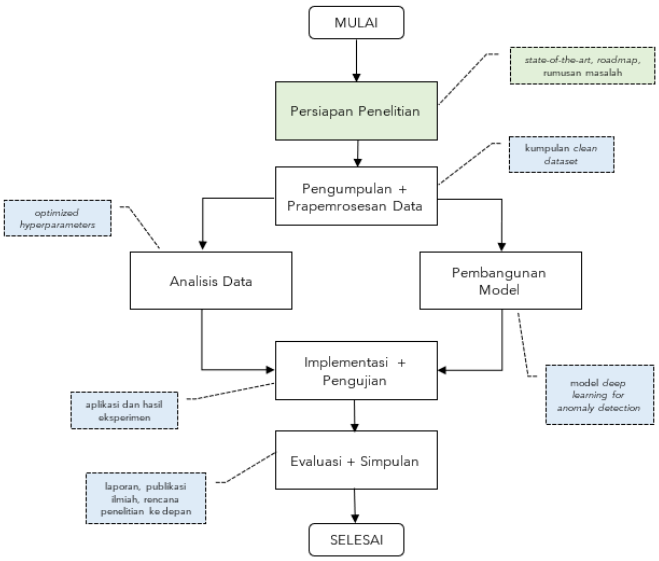
Amna Bamaqa mengusulkan penggunaan *Hierarchical Temporal Memory* (HTM) untuk melakukan deteksi anomali pada kerumunan. Dataset yang digunakan dibuat menggunakan MassMotion *crowd simulation tool*. Model yang diusukan mendapatkan nilai F_measure 94,6% [18]. Sivalingan H. mengusulkan sebuah teknik baru yang disebut *Video Swin Transformer (*VST)-*Anomaly Scoring Network* (VASN). Dataset yang digunakan merupakan Crowd Anomaly Detection UML Dataset dan mendapatkan Anomaly Score Value sebesar 98,92% [19].

Mehmood dan Jing Li mengusulkan penggunaan *Convolutional Neural Network* (CNN). Mereka menggunakan arsitektur yang berbeda untuk melakukan deteksi anomali di kerumunan. Mehmood menggunakan Xception dan menghasilkan akurasi hingga 99% pada dataset pertandingan hockey [20]. Jing Li melakukan eksperimen dengan dataset yang lebih beragam dan mencapai akurasi 97,1% [21].

Dari sisi komputasi, Pelati mengusulkan penggunaan *Long Short-Term Memory* (LSTM) untuk deteksi anomali. Tujuannya adalah model dengan biaya komputasi yang rendah. Tantangan yang muncul adalah akurasi yang kurang dari 80% [22]. Sabih mengusulkan penggunaan CNN untuk ekstraksi fitur dan LSTM untuk deteksi anomali, dengan menggabungkan 2 metode tersebut model berhasil mencapai nilai akurasi 92.2% [23].

Berdasarkan penelitian sebelumnya, penelitian ini mengusulkan sebuah metode dengan menambahkan lapisan untuk melakukan deteksi objek sebelum deteksi anomali. Metode yang dipilih untuk melakukan deteksi objek adalah *You Only Look Once v5* (YOLOv5), metode ini terbukti memiliki kemampuan yang baik dalam mendeteksi objek pada keramaian dan berhasil mendapatkan nilai mAP50 sebesar 98,5% pada penelitian [24]. Penelitian ini menggabungkan kemampuan YOLOv5 dan LSTM untuk mendeteksi anomali pada keramaian. LSTM dipilih atas kemampuannya dalam mengenali pola yang baik dengan biaya komputasi yang rendah [25] sehingga dapat digunakan di perangkat komputasi tepi. Penelitian ini berfokus untuk membandingkan hasil dari kombinasi *hyperparameter* dan mencari model yang paling optimal untuk digunakan dalam mendeteksi anomali pada keramaian.

2. METODE PENELITIAN



Gambar 1. Alur Penelitian

Pendekatan yang dirumuskan dalam penelitian ini dibagi ke dalam dua hal yaitu deteksi objek menggunakan YOLOv5 dan *object tracking* menggunakan LSTM. Gambar 1 menunjukkan alur

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4224

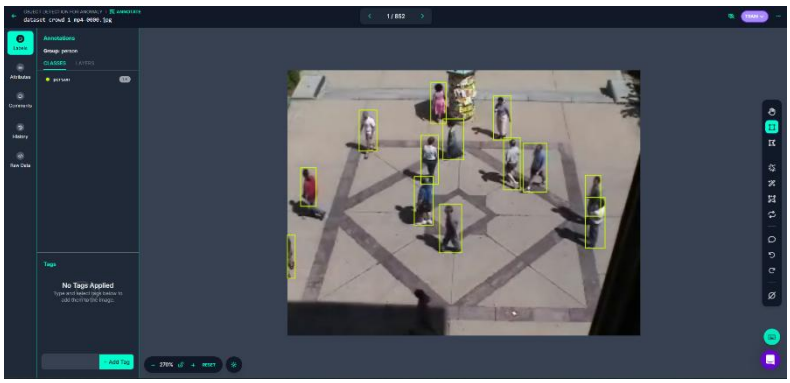
penelitian dalam implementasi YOLOv5 dan LSTM dalam melakukan deteksi anomali pada keramaian. Penelitian terdiri atas beberapa tahap. Tahap pertama dimulai dengan pengumpulan dataset dan melakukan prapemrosesan data citra. Kemudian dilanjutkan dengan analisis data untuk menentukan *hyperparameter* yang optimal dan membangun model YOLOv5 dan model LSTM untuk melakukan deteksi anomali. Selanjutnya kinerja dari model yang telah dibangun akan diuji menggunakan *confusion matrix*. Langkah terakhir adalah mengevaluasi hasil dan mengambil kesimpulan dari metode yang diusulkan pada penelitian ini.

2.1. Pengumpulan dan Prapemrosesan Data

Penelitian ini menggunakan cuplikan video dari Crowd Anomaly Detection UML Dataset yang berdurasi 1 menit 11 detik. Dataset ini diambil dari Kaggle yang berupa sebuah video kerumunan orang sedang bergerak normal dan kemudian berlari karena panik. Sebelum digunakan untuk pelatihan, data akan melalui beberapa tahap prapemrosesan. Pertama-tama data video akan diolah dengan proses ekstraksi *frame* untuk menghasilkan citra yang merepresentasikan setiap *frame*. Gambar 2 menunjukkan sampel *frame* pada dataset yang bersifat normal dan anomali.

Gambar 2. Sampel Dataset Normal dan Anomali.

Pada penelitian ini proses *frame extraction* dilakukan dengan bantuan alat dari Roboflow. Dataset diekstraksi menjadi 12 *frame* per detik sehingga menjadi 852 citra. Proses *image annotation* pada penelitian ini hanya terdapat 1 objek, maka hanya ada 1 kelas yaitu “Person”. Data yang di anotasikan merupakan seluruh orang yang terdapat di dalam *frame*. Setiap objek akan ditandai dengan cara memberikan *bounding box* sebagai pembatas objek dan latar. Gambar 3 menunjukkan contoh proses *image annotation*.



Gambar 3. Proses Image Annotation

Setelah proses *image annotation*, kita mendapatkan berkas teks yang berisikan label koordinat dari setiap objek yang terdapat pada citra. Gambar 4 menunjukkan contoh berkas teks yang didapatkan.

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 2, April 2025, Page. 951-964
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4224

Berkas teks berisikan label dengan 5 variabel yang dapat dilihat secara berurutan mulai dari kelas objek yang dideteksi, titik koordinat x objek, titik koordinat y objek, nilai lebar *bounding box* objek, dan nilai tinggi *bounding box* objek.

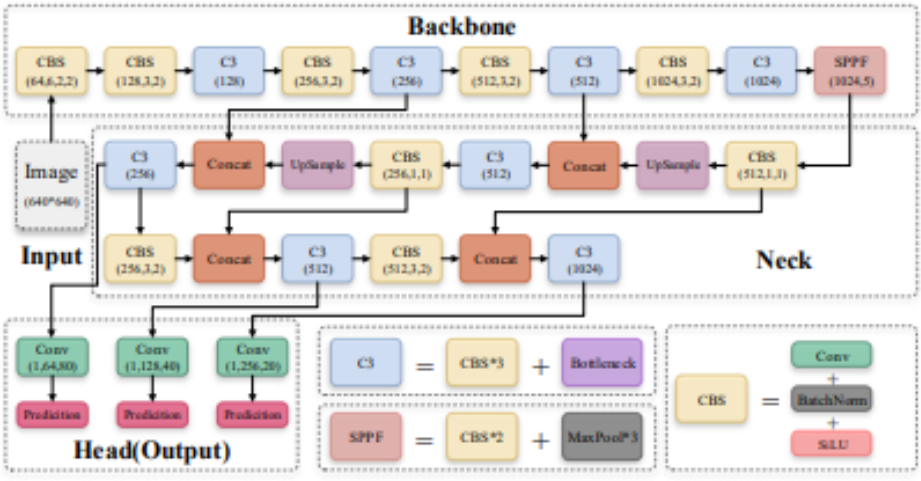
```
0 0.228125 0.22734375 0.05 0.15390625
0 0.403125 0.3375 0.05 0.18359375
0 0.7015625 0.36640625 0.065625 0.18359375
0 0.059375 0.43984375 0.04375 0.14609375
0 0.6359375 0.35234375 0.046875 0.19609375
0 0.609375 0.17890625 0.05 0.16640625
0 0.4625 0.59609375 0.0625 0.18359375
0 0.428125 0.12265625 0.04375 0.15390625
0 0.871875 0.575 0.05625 0.19140625
0 0.4703125 0.26015625 0.059375 0.15390625
0 0.3859375 0.49140625 0.053125 0.18359375
0 0.865625 0.47265625 0.04375 0.15390625
0 0.0109375 0.70234375 0.021875 0.1625
```

Gambar 4. Label Hasil Anotasi

Dataset dibagi menjadi 3 bagian yaitu data latihan, data validasi dan data uji dengan rasio 70% : 20% : 10%. Proses *splitting* ini menghasilkan 596 data latih, 171 data validasi, dan 85 data uji. Proses terakhir adalah menyamakan ukuran citra menjadi 640x640 piksel, ukuran ini dipilih karena cukup besar untuk menangkap detail penting dalam citra, sehingga model dapat mendeteksi objek dengan akurasi yang baik. Namun, ukurannya tidak terlalu besar, sehingga pemrosesan tetap cepat dibandingkan dengan resolusi yang lebih tinggi.

2.2. Pembangunan Model

Pada tahap ini, arsitektur dari model yang digunakan didefinisikan. Model yang digunakan untuk melakukan deteksi objek adalah YOLOv5. YOLOv5 merupakan sebuah arsitektur jaringan saraf konvolusional yang dirancang untuk melakukan deteksi objek. Arsitektur ini terdiri dari tiga bagian utama, yaitu *backbone*, *neck*, dan *head* (Gambar 5). *Backbone* adalah sebuah arsitektur CNN yang digunakan untuk melakukan ekstraksi fitur dari citra dengan berbagai ukuran. *Neck* adalah lapisan yang berfungsi untuk melakukan proses penggabungan fitur yang sebelumnya telah diekstraksi pada *backbone*. *Head* adalah lapisan terakhir yang digunakan untuk melakukan prediksi terhadap objek yang terdapat pada citra [26].



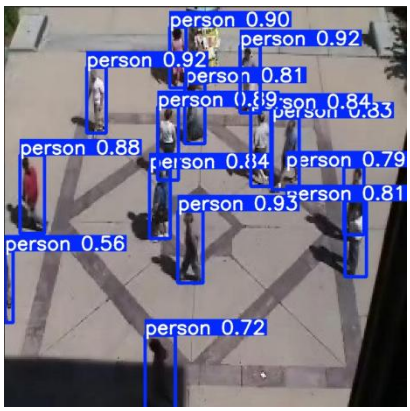
Gambar 5. Arsitektur YOLOv5

Arsitektur YOLOv5 yang terdiri atas beberapa modul dengan beberapa lapis proses. Modul CBS yang terdiri dari proses konvolusi, *Batch Normalization*, dan SiLU. Modul C3 (*Concentrated-*

Comprehensive Convolution) yang terdiri atas tiga lapisan konvolusi dan *bottleneck*. Terakhir Modul SPPF atau *Spatial Pyramid Pooling-Fast* yang terdiri atas dua lapisan CBS dan tiga lapisan *max pooling* [27].

Model yang digunakan adalah YOLOv5s. YOLOv5s merupakan model YOLOv5 dengan ukuran yang kecil dengan sekitar 7 juta parameter dan 16 GFLOPs sehingga memiliki waktu komputasi yang lebih cepat dibandingkan model-model YOLOv5 lainnya. Model tersebut akan dilatih sebanyak 9 kali dengan kombinasi hyperparameter yaitu, ukuran citra 640x640, batch 16, epoch 50/100/150, dan learning rate 0,01/0,001/0,0001.

Model YOLOv5 yang telah dilatih digunakan untuk mendeteksi data lokasi dari setiap objek yang dideteksi. Model akan menghasilkan citra dengan anotasi sesuai prediksi dari model dan file teks yang berisikan label dari objek. Gambar 6 menunjukkan hasil deteksi berupa citra dan teks.




```
0 0.0101562 0.695312 0.0203125 0.178125 0.55632
0 0.385938 0.907812 0.075 0.184375 0.718253
0 0.867188 0.48125 0.05 0.16875 0.791454
0 0.471875 0.263281 0.05 0.151562 0.8078
0 0.871875 0.575781 0.053125 0.189063 0.81489
0 0.697656 0.364062 0.0671875 0.178125 0.833512
0 0.385156 0.488281 0.0515625 0.173438 0.840738
0 0.635938 0.349219 0.05 0.192188 0.844902
0 0.0679687 0.464844 0.0609375 0.195312 0.880337
0 0.405469 0.338281 0.0515625 0.182812 0.886273
0 0.429688 0.124219 0.040625 0.151562 0.901305
0 0.227344 0.230469 0.0484375 0.164062 0.917369
0 0.609375 0.178906 0.05 0.167187 0.920484
0 0.460938 0.596094 0.0625 0.179688 0.925522
```

Gambar 6. Hasil Deteksi

Model-model YOLOv5 yang sudah dilatih dengan menggunakan beberapa kombinasi *hyperparameter* akan digunakan untuk deteksi objek pada dataset. Hasil dari tahap ini adalah dataset citra yang sudah dianotasi oleh model dan berkas teks berisikan informasi dari objek yang terdeteksi. Berkas teks akan diolah menjadi berkas *comma separated value* (CSV) yang berisikan vektor pergerakan x dan y dari masing masing objek dimana setiap baris mewakili perpindahan antar *frame* (baris 1 menggambarkan perpindahan objek dari *frame* 1 ke *frame* 2). Pertama-tama setiap objek akan ditentukan pasangan antar *frame*-nya dengan menggunakan metode *Nearest-neighbour*. Setelah pasangan setiap objek ditemukan, hitung jarak perpindahan antar objek menggunakan persamaan *Euclidian Distance* hasilnya disimpan dalam vektor perubahan. Langkah selanjutnya adalah menambahkan label untuk menandai *frame* yang terdapat anomali di dalamnya. *Frame* dengan anomali akan diberi label 1 dan *frame* normal akan diberi label 0. Tabel 1 menunjukkan contoh data yang akan digunakan untuk pelatihan LSTM.

Tabel 1. Data vektor pergerakan

Person1 x	Person1 y	Person2 x	Person2 y	…	Person9 x	Person9 y	Label
10	14	-10	-12	…	8	-9	0
6	4	-7	-9	…	-10	-6	1
…	…	…	…	…	…	…	…
5	8	-3	7	…	10	11	0

Karena jumlah objek yang dideteksi pada setiap *frame* berbeda-beda, langkah selanjutnya adalah mengisi nilai kosong menjadi -999 yang berfungsi untuk melakukan proses *masking* sehingga nilai kosong tidak mempengaruhi hasil dari pelatihan model.

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4224

Selanjutnya dibangun model LSTM untuk deteksi anomali. LSTM merupakan peningkatan model dari *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi permasalahan *vanishing* dan *exploding gradient* [28]. LSTM memiliki dua jenis lapisan yaitu *state* yang bertugas sebagai memori untuk menyimpan informasi yang akan digunakan pada lapisan berikutnya dan *gate* yang bertugas untuk menentukan informasi mana yang harus diingat atau dilupakan. LSTM terdiri dari tiga mekanisme utama yaitu [29]:

- *Output control* adalah jumlah *neuron* yang berpengaruh pada keluaran sebelumnya dan keadaan saat ini.
- *Memory control* adalah jumlah dari keadaan sebelumnya yang akan dihapus atau dilupakan pada keadaan saat ini.
- *Input control* adalah jumlah dari keluaran sebelumnya dan keadaan saat ini yang akan dipertimbangkan untuk keadaan selanjutnya.

Model LSTM yang akan digunakan terdiri atas 3 lapisan yaitu, lapisan *masking*, lapisan LSTM, dan lapisan *dense* dengan fungsi aktivasi RelU. Pelatihan akan dilakukan sebanyak 81 kali untuk mendapatkan kombinasi *hyperparameter* yang terbaik. Nilai *hyperparameter* yang akan diuji yaitu, *epoch* dengan nilai 50/100/150 dan LSTM unit dengan nilai 32/64/128. Masing-masing kombinasi akan dilatih dengan 9 dataset berbeda yang didapatkan dari hasil deteksi 9 model YOLOv5 yang telah dilatih. Tabel 2 menunjukkan arsitektur yang digunakan pada model ini.

Tabel 2. Arsitektur Model LSTM

Lapisan	Deskripsi	Keluaran
Masking	Mengabaikan nilai *padding* (-999)	(None, timesteps, features)
LSTM	x unit LSTM	(None, x)
Dense	Lapisan *dense* dengan fungsi aktivasi ReLU	(None, 1)

2.3. Implementasi dan Pengujian

Pada tahap ini, model yang sudah dilatih akan diuji menggunakan *confusion matrix* untuk menentukan model mana yang memiliki hasil terbaik. *Confusion matrix* merupakan sebuah matriks yang menunjukkan hasil klasifikasi prediksi dan klasifikasi yang sebenarnya [30].

Tabel 3. *Confusion Matrix*

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Tabel 3 menunjukkan *confusion matrix* mempunyai empat klasifikasi, yaitu :

- TP (*True Positive*) merupakan jumlah sampel positif yang diklasifikasikan sebagai positif.
- FP (*False Positive*) merupakan jumlah sampel negatif yang diklasifikasikan sebagai positif.
- TN (*True Negative*) merupakan jumlah sampel negatif yang diklasifikasikan sebagai negatif.
- FN (*False Negative*) merupakan jumlah sampel positif yang diklasifikasikan sebagai negatif.

Setelah mendapatkan *confusion matrix* kita dapat menghitung nilai precision, recall, dan mAP untuk menilai kinerja dari model YOLOv5 dan accuracy, precision dan F1-Score untuk menilai kinerja dari model LSTM.

$$Recall = \frac{TP}{TP+FN} \quad (1)$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} APi \quad (2)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

Recall merupakan matriks yang menilai kemampuan model untuk menemukan semua objek yang seharusnya positif. *Recall* menghitung persentase dari objek yang diidentifikasikan secara benar sebagai positif dari seluruh objek yang seharusnya diidentifikasi sebagai positif [31]. Persamaan 1 menunjukkan cara menghitung nilai *recall*. *Mean Average Precision* (mAP) merupakan matriks evaluasi yang biasa digunakan untuk menilai kemampuan model dalam mendeteksi objek, mAP menghitung rata-rata antara *precision* dan *recall* [32]. Persamaan 2 menunjukkan cara menghitung nilai mAP, Dimana N merupakan jumlah total kelas dalam data, i merupakan indeks, dan APi merupakan *Average Precision* untuk kelas ke-i. *Accuracy* merupakan matriks yang biasa digunakan dalam *machine learning* untuk menilai akurasi dari model. *Accuracy* menghitung persentase dari seluruh objek yang diidentifikasikan secara benar. Persamaan 3 menunjukkan cara menghitung nilai *accuracy*.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

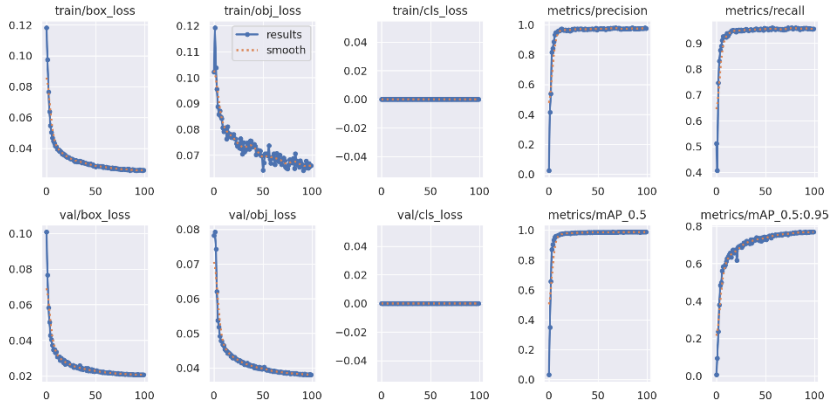
$$F1\ Score = \frac{2\ x\ Precision\ x\ Recall}{Precision\ x\ Recall} \quad (5)$$

Precision merupakan matriks yang biasa digunakan dalam deteksi objek untuk menilai akurasi dari model. *Precision* menghitung persentase dari objek yang diidentifikasikan secara benar sebagai positif dari seluruh objek yang diidentifikasi sebagai positif [33]. Persamaan 4 menunjukkan cara menghitung nilai *precision*. F1-Score merupakan matriks yang digunakan untuk menilai kinerja model secara keseluruhan. Nilai dari F1-Score didapatkan dari rata-rata dari *precision* dan *recall*. Persamaan 5 menunjukkan cara menghitung nilai F1-Score.

3. HASIL DAN PEMBAHASAN

Tabel 4 menunjukkan 10 contoh model dengan hasil terbaik. Model YOLOv5 memiliki nilai mAP50 lebih besar, cenderung mendapatkan nilai akurasi yang tinggi juga pada saat melakukan deteksi anomali. Model 5 dan 8 memiliki hasil deteksi anomali yang serupa, akan tetapi model 5 memiliki waktu komputasi yang lebih cepat. Model yang paling optimal adalah model 5.2 dan 5.3. Model 5.2 memiliki keunggulan dengan nilai F1-Score yang sangat tinggi sehingga dapat meminimalisir kejadian anomali yang tidak terdeteksi, akan tetapi model 5.3 memiliki nilai *precision* yang sempurna sehingga meminimalisir peringatan yang keliru. Secara keseluruhan dapat disimpulkan bahwa model YOLOv5 dengan nilai *epoch* 100 dan *learning rate* 0,001 mendapatkan hasil terbaik secara keseluruhan dalam mendeteksi anomali dengan waktu yang cukup baik.

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 2, April 2025, Page. 951-964
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4224



Gambar 7. Hasil Pelatihan Model YOLOv5 ke 5

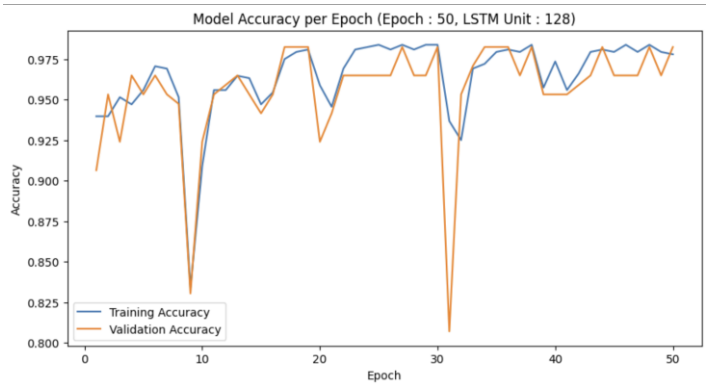
Gambar 7 menunjukkan hasil pelatihan dari model YOLOv5 untuk melakukan deteksi objek. Nilai dari train/box_loss pada *epoch* pertama hingga ke 100 menurun dari 0,12 hingga di bawah 0,03 dan nilai val/box_loss dari 0,10 hingga 0,02 yang menandakan kemampuan model untuk menentukan lokasi objek meningkat. Nilai train/obj_loss menurun dari 0,12 hingga di bawah 0,07 dan nilai val/obj_loss menurun dari 0,08 hingga 0,04, hal ini menandakan model belajar dengan baik untuk mendeteksi objek pada citra. Nilai dari train/cls_loss dan val/cls_loss berada di 0 dikarenakan hanya ada satu kelas yang dideteksi pada penelitian ini. Model ini berhasil mendapatkan nilai *precision* 0,975, *recall* 0,959, mAP50 0,988 dan mAP95 0,769.

Tabel 4. Perbandingan Hasil Pelatihan Model

No	YOLOv5		LSTM		Hasil YOLOv5			Hasil LSTM		
	Epoch	Learning Rate	Epoch	LSTM Unit	Time	Recall	mAP 50	Accuracy	Precision	F1-Score
1.4	50	0.01	100	32	4m 26d	0.968	0.987	0.965	0.733	0.786
1.5	50	0.01	100	64	4m 26d	0.968	0.987	0.953	0.632	0.75
3.1	50	0.0001	50	32	5m 2d	0.919	0.964	0.947	1	0.471
5.2	100	0.001	50	64	8m 42d	0.959	0.988	0.953	0.692	0.962
5.3	100	0.001	50	128	8m 42d	0.959	0.988	0.982	1	0.869
6.2	100	0.0001	50	64	9m 25d	0.944	0.975	0.965	1	0.7
7.8	150	0.01	150	64	12m 50d	0.962	0.984	0.982	1	0.869
8.1	150	0.001	50	32	13m 8d	0.96	0.989	0.982	1	0.869
8.4	150	0.001	100	32	13m 8d	0.96	0.989	0.953	0.692	0.962
9.1	150	0.0001	50	32	13m 48d	0.946	0.979	0.97	1	0.761

Tabel 4 menunjukkan hasil pelatihan dari model 5.3, 5.7, 7.8, dan 8.1 berhasil mendapatkan nilai akurasi yang cukup tinggi di 0.982, dengan nilai precision 1 dan F1-Score yang cukup tinggi yaitu 0,869. Pengaruh nilai *epoch* pada model YOLOv5 berbanding lurus, ketika nilai *epoch* bertambah maka nilai dari akurasi pun bertambah, Hal ini disebabkan semakin banyak model belajar, semakin baik pula model dapat mendeteksi objek pada *frame* sehingga fitur yang dihasilkan lebih baik. Tetapi pengaruh nilai *epoch* pada model LSTM berbanding terbalik, ketika nilai *epoch* menurun maka nilai akurasi bertambah, hal ini disebabkan oleh terjadinya *overfitting* pada nilai *epoch* yang tinggi. Gambar 8 menunjukkan hasil pelatihan dari model 5.3, akurasi pada data pelatihan dan validasi cenderung tinggi di atas 0.9 akan tetapi terdapat fluktuasi di sekitar *epoch* 10 dan 30, hal ini menunjukkan potensi *overfitting* pada beberapa tahap pelatihan. *Overfitting* dapat disebabkan karena data pelatihan yang terlalu sedikit, untuk mengatasi hal ini dapat dilakukan proses augmentasi data atau penelitian dengan dataset yang memiliki ukuran

lebih besar dan pembagian kelas yang seimbang. Pada *epoch* 40 hingga 50 grafik cenderung cukup stabil, hal ini menunjukkan model sudah belajar dengan baik dari *epoch* ke 40.



Gambar 8. Hasil Pelatihan Model LSTM Terbaik

Pada kombinasi di model 5.2, model berhasil mendapatkan F1-Score yang tinggi yaitu 0,962 dan nilai akurasi sebesar 0,953, tetapi pada model ini nilai *Precision* cenderung kecil yaitu 0,692. Beberapa model lain juga yang berhasil mencapai nilai *Precision* yang sempurna yaitu 1, dengan nilai F1-Score yang bervariasi, seperti pada kombinasi model 3.1 dimana model berhasil mendapatkan nilai *precision* sempurna akan tetapi nilai F1-Score yang didapatkan sangat rendah di nilai 0,471. Beberapa model menghasilkan nilai 0 pada *Precision* dan F1-Score dikarenakan model tidak mendeteksi adanya *frame* anomali pada saat pengujian. Hal ini dapat disebabkan karena pembagian data yang tidak seimbang sehingga terjadi *overfitting* dimana model hanya memprediksi kelas mayoritas (*frame* normal) dan mengabaikan kelas minoritas (*frame* anomali).

Tabel 5. Perbandingan Metode

Method	Accuracy	Precision	F1-Score
Abdullah et al. [8]	0.943	0.941	0.941
Li et al. [17]	-	0.63	0.59
A. Mehmood [20]	0.976	0.972	-
Ours (YOLOv5 & LSTM – 5.2)	0.953	0.692	**0.962**
Ours (YOLOv5 & LSTM – 5.3)	**0.982**	1	0.869

Tabel 5 menunjukkan perbandingan hasil prediksi model yang dibangun dengan penelitian sebelumnya yang dalam mendeteksi anomali. Model 5.3 berhasil memiliki nilai akurasi dan presisi yang paling tinggi dibandingkan dengan penelitian sebelumnya, model tersebut berhasil meningkatkan nilai akurasi sebesar 1% dan nilai presisi sebesar 3% dibandingkan dengan penelitian [20]. Selain itu model 5.2 berhasil meningkatkan nilai F1-Score sebesar 2% dibandingkan dengan penelitian [8]. Hal ini menunjukkan kemampuan yang baik dari model yang dibangun dalam mendeteksi anomali.

4. DISKUSI

Dari penelitian ini dapat disimpulkan bahwa kombinasi model *You Only Look Once v5* (YOLOv5) dengan *Long Short-Term Memory* (LSTM) dapat memberikan kinerja yang sangat baik untuk mendeteksi anomali pada keramaian, penelitian ini berhasil mendapatkan akurasi senilai 98%, precision 100%, dan F1-Score 86% pada kombinasi *hyperparameter* pada YOLOv5 (*epoch* 100, *learning rate* 0,001) dan LSTM (*epoch* 50, LSTM *unit* 128). Hasil ini menandakan bahwa pendekatan ini mampu memanfaatkan kekuatan deteksi objek dan analisis temporal secara efektif. Jika dibandingkan dengan penelitian sebelumnya, pendekatan ini menunjukkan keunggulan yang signifikan dalam hal akurasi dan efisiensi. Sebagai contoh, penelitian [17] menggunakan strategi multi-stage clustering untuk mendeteksi

anomali dengan akurasi 87%. Meskipun menjanjikan, pendekatan ini menghadapi tantangan besar dalam biaya komputasi untuk aplikasi real-time, yang membatasi penggunaannya dalam skenario praktis.

Pendekatan berbasis *Hierarchical Temporal Memory* (HTM) yang diajukan pada penelitian [18] mampu mencapai F_measure sebesar 94,6%, namun hasil ini dicapai pada dataset simulasi keramaian yang dibuat menggunakan MassMotion yang mungkin tidak sepenuhnya mencerminkan kompleksitas dunia nyata. Penelitian berbasis deep learning juga menunjukkan hasil yang beragam. Penelitian [19] menggunakan *Convolutional Neural Network (CNN)* dengan arsitektur Xception dan mencapai akurasi 99% pada dataset pertandingan hockey. Namun, kinerja ini sangat bergantung pada dataset yang spesifik dan terbatas. Di sisi lain penelitian [20] mencapai akurasi 97,1% dengan menggunakan dataset yang lebih beragam, yang menunjukkan pentingnya kualitas dan representasi dari dataset yang digunakan dalam menentukan keberhasilan model deteksi anomali. Sementara itu, penelitian [21] mengeksplorasi potensi penggunaan LSTM untuk deteksi anomali. Pendekatan ini bertujuan untuk mendukung aplikasi pada perangkat komputasi tepi dengan konsumsi daya yang rendah. Namun, Pelati hanya mencapai akurasi di bawah 80%, yang menunjukkan bahwa LSTM sebagai model tunggal memiliki keterbatasan dalam menangani pola keramaian yang kompleks.

Kombinasi YOLOv5 dan LSTM dalam penelitian ini tidak hanya mengatasi keterbatasan beberapa metode sebelumnya, tetapi juga memberikan pendekatan baru yang efisien untuk analisis spasial dan temporal. Meskipun demikian, tantangan seperti ketidakseimbangan kelas dalam dataset yang menyebabkan model *overfitting* masih perlu diatasi. Hal ini penting untuk memastikan model dapat generalisasi dengan baik pada berbagai kondisi dunia nyata. Penelitian selanjutnya dapat berfokus pada optimasi dataset, seperti balancing data menggunakan *oversampling* atau teknik *loss function* khusus. Selain itu, eksplorasi arsitektur *hybrid*, misalnya dengan menambahkan mekanisme *Attention* pada LSTM, dapat membantu menangkap pola keramaian yang lebih kompleks. Implementasi real-time pada perangkat komputasi tepi juga menjadi tantangan menarik untuk ditangani di masa depan.

5. KESIMPULAN

Penelitian ini bertujuan untuk membandingkan hasil dari kombinasi *hyperparameter* dan mencari model yang paling optimal untuk mendeteksi anomali pada komputasi tepi. Penelitian dibagi ke dalam 2 bagian, yaitu *object detection* dan *object tracking.* Penelitian dilakukan menggunakan metode YOLOv5 untuk *object detection* dan LSTM untuk *object tracking.* Kedua hal tersebut digabungkan untuk mendeteksi anomali pada keramaian, menggunakan dataset video berdurasi 1 menit 11 detik.

Eksperimen dilakukan dengan menggunakan beberapa *hyperparameter* pada kedua metode. Pada YOLOv5 *hyperparameter* yang digunakan adalah *epoch* dengan nilai 50/100/150 dan *learning rate* dengan nilai 0,01/0,001/0,0001. Pada LSTM *hyperparameter* yang digunakan adalah *epoch* dengan nilai 50/100/150 dan LSTM unit dengan nilai 32/64/128. Kombinasi model YOLOv5 dengan konfigurasi *epoch* 100, dan *learning rate* 0,001 dan model LSTM dengan konfigurasi *epoch* 150 dan jumlah LSTM unit sebanyak 32 berhasil mendapatkan nilai akurasi tertinggi yaitu 98% dengan waktu komputasi terendah. Sementara itu kombinasi model YOLOv5 dengan konfigurasi *epoch* 50, dan *learning rate* 0.01 dan model LSTM dengan konfigurasi *epoch* 150 dan jumlah LSTM unit sebanyak 32 berhasil mendapatkan nilai *precision* tertinggi yaitu 100% dengan waktu komputasi terendah. Kemudian kombinasi model YOLOv5 dengan konfigurasi *epoch* 100, dan *learning rate* 0,001 dan model LSTM dengan konfigurasi *epoch* 50 dan jumlah LSTM unit sebanyak 64 berhasil mendapatkan nilai F1-Score tertinggi yaitu 96% dengan waktu komputasi terendah.

Penelitian ini memiliki keterbatasan pada variasi dataset yang digunakan, terdapat ketidakseimbangan kelas pada dataset dimana *frame* anomali yang berjumlah 55 *frame* berjumlah jauh lebih sedikit dibandingkan *frame* normal yang berjumlah 797 *frame*. Dataset yang digunakan juga

memiliki konteks yang terbatas. Penelitian ini memiliki implikasi dalam meningkatkan efisiensi deteksi anomali pada keramaian di komputasi tepi. Untuk meningkatkan kembali kinerja pada penelitian selanjutnya, ada beberapa saran yang mungkin dapat menghasilkan hasil yang maksimal, yaitu dengan menggunakan dataset yang lebih beragam, misalnya: data dalam industri manufaktur, data kejadian tindak kriminal, atau manusia hanyut di tepi pantai. Saran lainnya adalah penggunaan model *deep learning* untuk *time series data* seperti GRU dan Bi-LSTM untuk mampu mendeteksi anomali dengan kualitas video yang minimalis atau pencahayaannya kurang.

DAFTAR PUSTAKA

[1] Fernández-Ares, P. García-Sánchez, M. G. Arenas, A. M. Mora and P. A. CastilloValdivieso, "Detection and Analysis of Anomalies in People Density and Mobility Through Wireless Smartphone Tracking," IEEE Access, vol. 8, pp. 54237-54253, 2020, doi: 10.1109/ACCESS.2020.2979367.

[2] T. Kulshrestha, D. Saxena, R. Niyogi and J. Cao, "Real-Time Crowd Monitoring Using Seamless Indoor-Outdoor Localization," IEEE Transactions on Mobile Computing, vol. 19, no. 3, pp. 664-679, 2020, doi: 10.1109/TMC.2019.2897561.

[3] E. I. Sari, " Angka Kematian Kecelakaan Transportasi Turun Sepanjang 2023." 2023. https://indonesia.go.id/kategori/editorial/7879/angka-kematian kecelakaantransportasi-turun-sepanjang-2023 (accessed Mar. 11, 2024).

[4] Kementerian Ketenagakerjaan Republik Indonesia, "Kecelakaan kerja di Indonesia Tahun 2023." 2023. https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fsatudata.kemnaker.go.id%2Fsatudatapublic%2F2023%2F11%2Ffiles%2Fdata%2F1708925833582_Jumlah%252520kecelakaan%252520kerja%252520tahun%2525202023.xlsx&wdOrigin=BROWSELINK (accessed Mar. 12, 2024).

[5] X. Zhang, J. Fang, B. Yang, S. Chen and B. Li, "Hybrid Attention and Motion Constraint for Anomaly Detection in Crowded Scenes," IEEE Transactions on Circuits and Systems for Video Technology, vol. 33, no. 5, pp. 2259-2274, 2023, doi: 10.1109/TCSVT.2022.3221622.

[6] Z. Ilyas, Z. Aziz, T. Qasim, et al., "A hybrid deep network based approach for crowd anomaly detection," *Multimedia Tools and Applications*, vol. 80, pp. 24053–24067, 2021, doi: 10.1007/s11042-021-10785-4.

[7] C. Direkoglu, "Abnormal Crowd Behavior Detection Using Motion Information Images and Convolutional Neural Networks," IEEE Access, vol. 8, pp. 80408–80416, 2020, doi: 10.1109/ACCESS.2020.2990355.

[8] F. Abdullah, M. Abdelhaq, R. Alsaqour, M. H. Alatiyyah, K. Alnowaiser, S. S. Alotaibi, *et al*., "Context Aware Crowd Tracking and Anomaly Detection via Deep Learning and Social Force Model," *IEEE Access*, vol. 11, pp. 75884-75898, 2023, doi: 10.1109/ACCESS.2023.3293537.

[9] A. A. Khan, M. A. Nauman, M. Shoaib, R. Jahangir, R. Alroobaea, M. Alsafyani, et al., "Crowd Anomaly Detection in Video Frames Using Fine-Tuned AlexNet Model," Electronics, vol. 11, no. 19, p. 3105, 2022, doi: 10.3390/electronics11193105.

[10] K. Pawar and V. Attar, "Application of Deep Learning for Crowd Anomaly Detection from Surveillance Videos," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 506-511, 2021, doi: 10.1109/Confluence51648.2021.9377055.

[11] C.-W. Chang, C. Chang, and Y.-Y. Lin, "A hybrid CNN and LSTM-based deep learning model for abnormal behavior detection," Multimedia Tools and Applications, vol. 81, pp. 11825–11843, Jan. 2022, doi: 10.1007/s11042-021-11887-9.

[12] A. Arif and A. Jalal, ''Automated body parts estimation and detection using salient maps and Gaussian matrix model,'' in Proc. Int. Bhurban Conf. Appl. Sci. Technol. (IBCAST), 2021, pp. 667–672, doi: 10.1109/IBCAST51254.2021.9393268.

[13] K. Rezaee, S. M. Rezakhani, M. R. Khosravi, and M. K. Moghimi, ''A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance,'' Pers.

Ubiquitous Comput., 2021, vol. 28, pp. 135–151, doi: 10.1007/s00779-021-01586-5.
[14] K. Boekhoudt, A. Matei, M. Aghaei, and E. Talavera, ‘‘HR-Crime: Humanrelated anomaly detection in surveillance videos,’’ in Proc. CAIP. Cham, Switzerland: Springer, 2021, vol. 13053, pp. 164–174, doi: 10.1007/978-3-030-89131-2_15.
[15] L. Xia and Z. Li, "A new method of abnormal behavior detection using LSTM network with temporal attention mechanism," The Journal of Supercomputing, vol. 77, no. 4, pp. 3223–3241, 2021, doi: 10.1007/s11227-020-03391-y.
[16] W. Luo, W. Liu, D. Lian, and S. Gao, "Video anomaly detection with sparse coding inspired deep neural networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 3, pp. 1070–1084, 2021, doi: 10.1109/TPAMI.2019.2944377.
[17] X. Li, M. Chen and Q. Wang, "Quantifying and Detecting Collective Motion in Crowd Scenes," IEEE Transactions on Image Processing, vol. 29, pp. 5571-5583, 2020, doi: 10.1109/TIP.2020.2985284.
[18] A. Bamaqa, M. Sedky, and B. Bastaki, "Reactive and Proactive Anomaly Detection in Crowd Management Using Hierarchical Temporal Memory," *International Journal of Machine Learning and Computing*, vol. 12, no. 1, pp. 7-16, 2022, doi: 10.18178/ijmlc.2022.12.1.1072.
[19] H. Sivalingan and N. Anandakrishnan, "Crowd Localization and Anomaly Detection Using Video Anomaly Scoring Network," *MSEA*, vol. 72, no. 1, pp. 825–837, Mar. 2023, doi: 10.17762/msea.v72i1.2055.
[20] A. Mehmood, "Efficient Anomaly Detection in Crowd Videos Using Pre-Trained 2D Convolutional Neural Networks," IEEE Access, vol. 9, pp. 138283-138295, 2021, doi: 10.1109/ACCESS.2021.3118009.
[21] J. Li, Q. Huang, Y. Du, X. Zhen, S. Chen and L. Shao, "Variational Abnormal Behavior Detection With Motion Consistency," IEEE Transactions on Image Processing, vol.31, pp. 275-286, 2022, doi: 10.1109/TIP.2021.3130545.
[22] A. Pelati, M. Meo and P. Dini, "Traffic Anomaly Detection Using Deep SemiSupervised Learning at the Mobile Edge," IEEE Transactions on Vehicular Technology, vol. 71, no. 8, pp. 8919-8932, Aug. 2022, doi: 10.1109/TVT.2022.3174735.
[23] M. Sabih and D. K. Vishwakarma, "Crowd anomaly detection with LSTMs using optical features and domain knowledge for improved inferring," *The Visual Computer*, vol. 38, pp. 1719–1730, 2022, doi: 10.1007/s00371-021-02100-x.
[24] M. Ş. Gündüz and G. Işık, "A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models," *Journal of Real-Time Image Processing*, vol. 20, no. 5, pp. 1239–1254, 2023, doi: 10.1007/s11554-023-01276-w.
[25] T. Saba, "Real-time anomalies detection in crowd using convolutional long short-term memory network," *Journal of Information Science*, vol. 49, no. 5, pp. 1145-1152, 2023, doi: 10.1177/01655515211022665.
[26] G. Jocher, ”Ultralytics YOLOv5 Architecture,” 2023. https://docs.ultralytics.com/yolov5/tutorials/architecture_description/ (accessed Mar. 23, 2024).
[27] H. Liu, F. Sun, J. Gu, and L. Deng, "SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode," *Sensors*, vol. 22, no. 15, pp. 5817, Aug. 2022, doi: 10.3390/s22155817.
[28] A.N. Moustafa and W. Gomaa, "Gate and common pathway detection in crowd scenes and anomaly detection using motion units and LSTM predictive models," Multimed Tools Appl., vol. 79, pp. 20689–20728, 2020. doi: 10.1007/s11042-020-08840-7.
[29] P. Rivas, Deep Learning for Beginners: A beginner’s guide to getting up and running with deep learning from scratch using Python. Packt Publishing Ltd.
[30] I. P. Sary, S. Andromeda, and E. U. Armin, “Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images,” Ultima Computing: Jurnal Sistem Komputer, vol. 15, no. 1, pp. 8– 13, 2023, doi: 10.31937/sk.v15i1.3204.
[31] U. Nepal and H. Eslamiat, “Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs,” Sensors, vol. 22, no. 2, 2022, doi: 10.3390/s22020464.
[32] K. Khairunnas, E. M. Yuniarno, and A. Zaini, “Pembuatan Modul Deteksi Objek Manusia

Menggunakan Metode YOLO untuk Mobile Robot," Jurnal Teknik ITS, vol. 10, no. 1, pp. A50–A55, 2021, doi: 10.12962/j23373539.v10i1.61622.

[33] E. Casas, L. Ramos, E. Bendek, and F. RivasEcheverria, "Assessing the Effectiveness of YOLO Architectures for Smoke and Wildfire Detection," IEEE Access, vol. 11, pp. 96554– 96583, 2023, doi: 10.1109/ACCESS.2023.3312217.

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 2, April 2025, Page. 951-964
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4224