

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Beberapa penelitian sebelumnya telah berkontribusi signifikan dalam pengembangan sistem deteksi jatuh, terutama dengan penerapan *Convolutional Neural Network* (CNN) dan metode berbasis visi. Berikut adalah beberapa penelitian sejenis beserta temuan dan hasilnya dapat dilihat pada tabel 2.1.

Tabel 2. 1 Tabel Penelitian Terdahulu

Peneliti	Deskripsi	Tujuan	Temuan	Kelebihan	Kekurangan
Alanazi dan Muhammadi (2022)	Mengembangkan model deteksi jatuh menggunakan penggabungan beberapa aliran data gambar dari video. Menggunakan dataset Le2i dengan 6.392 urutan video.	Meningkatkan akurasi deteksi jatuh real-time dengan pendekatan multi-stream	Akurasi: 99.03% Sensitivitas: 99.00% Spesifisitas: 99.68% Presisi: 99.00% Performa lebih baik dari ResNet dan GoogleNet	Akurasi sangat tinggi Sensitivitas dan spesifisitas seimbang Performa real-time superior	Membutuhkan dataset besar Kompleksitas komputasi tinggi Proses image fusion kompleks
Kong et al. (2019)	Mengeksplorasi penggunaan representasi spatiotemporal untuk deteksi jatuh melalui video pengawasan dengan memanfaatkan optical flow	Memanfaatkan informasi gerakan untuk meningkatkan akurasi deteksi jatuh dalam video pengawasan	Berhasil mendeteksi jatuh secara efisien di lingkungan nyata dengan memanfaatkan informasi gerakan	Ekstraksi informasi gerakan detail Efektif di lingkungan nyata Analisis temporal yang baik	Bergantung pada kualitas optical flow Sensitif terhadap pencahayaan Kompleksitas komputasi tinggi

Casilari et al. (2020)	Mengevaluasi kinerja CNN pada berbagai dataset publik untuk mendeteksi jatuh dalam kondisi yang berbeda	Mengevaluasi efektivitas CNN pada berbagai dataset dan kondisi yang berbeda	CNN memberikan hasil baik jika dioptimalkan untuk dataset tertentu, namun menghadapi tantangan dalam generalisasi	Evaluasi komprehensif Pemahaman mendalam tentang limitasi Identifikasi kebutuhan optimasi	Kesulitan generalisasi Performa tidak konsisten Butuh penyesuaian per kasus
Harrou et al. (2018)	Mengembangkan sistem deteksi jatuh berbasis visi menggunakan kamera kedalaman untuk monitoring lansia	Meningkatkan keamanan lansia dengan sistem deteksi jatuh yang akurat	Penggunaan data kedalaman berhasil mengurangi false positives dan meningkatkan reliabilitas	Pengurangan false positive Cocok untuk monitoring lansia Deteksi akurat dengan data 3D	Ketergantungan pada depth camera Jangkauan terbatas Biaya implementasi tinggi
Huang et al. (2018)	Mengembangkan sistem deteksi jatuh menggunakan pose estimation untuk mengidentifikasi perubahan postur tubuh	Mengembangkan sistem peringatan otomatis berbasis pose estimation	Sistem berhasil mengenali aktivitas berisiko dan memicu peringatan otomatis	Deteksi perubahan postur Sistem peringatan otomatis Monitoring real-time	Bergantung pada kualitas pose estimation Kesulitan pada pose kompleks Terdampak oklusi

2.2 Computer Vision

Computer Vision merupakan salah satu dari *Artificial Intelligence* yang memiliki fokus pada perkembangan dari teknologi dan algoritma pada komputer agar dapat melihat, memproses, memahami dan menganalisis data visual berupa gambar dan video untuk dapat mengetahui dan memproses informasi yang diterima berupa tiga dimensi seperti yang dilakukan oleh manusia. *Computer Vision* adalah bidang yang dominan untuk digunakan pada penelitian dan serbaguna di era saat ini. *Computer Vision* memberikan intruksi padakomputer untuk dapat memahami dan menganalisis dengan pemahaman tingkat tinggi mengenai konten visual. Sub bidang dari *computer vision* meliputi pengenalan pada adegan atau objek, deteksi objek, pelacakan video, segmentasi objek, pose dan estimasi gerakan, pemodelan pemandangan, dan restorasi gambar (Morris dalam Diwan et al., 2023).

Computer vision memiliki kemampuan untuk dapat mengembangkan sistem secara otomatis dengan melihat, mengenali, dan menganalisis gambar atau video. Tujuan utama yang dimiliki oleh *computer vision* adalah agar komputer dapat melihat dan memahami dunia nyata yang berupa visual layaknya manusia, dengan menggunakan algoritma dan teknik dalam melakukan proses gambar agar dapat dianalisis dan mendapatkan informasi yang berguna (Sutisna, 2024). Mempelajari metode agar mendapatkan informasi numerik maupun simbolik merupakan bidang ilmu dari *Computer Vision*. Proses-proses yang dapat dilakukan oleh *computer vision* sangat beragam, seperti dengan melakukan penangkapan citra, peningkatan citra, segmentasi, ekstraksi pada fitur, serta pada melakukan klasifikasi pada gambar atau video. Dengan sebagian besar tugas dari *computer vision* itu senidiri adalah mendapatkan informasi mengenai peristiwa atau suatu deskripsi yang berasal dari gambar digital dan juga ekstraksi pada fitur. Dalam penggunaannya, *computer vision* dapat memecahkan masalah mengenai informasi yang akan dianalisisnya sesuai dengan domain aplikasi dan sifat dari datayang akan dianalisis (Wiley & Lucas, 2018).

2.3 Deep Learning

Deep Learning atau biasa disebut dengan istilah Pembelajaran Struktural Mendalam (*Deep Structured Learning*) atau Pembelajaran Hierarki (*Hierarchical learning*) merupakan salah satu cabang dari ilmu *Machine Learning*, yaitu algoritma pemodelan abstraksi pada tingkat tinggi yang ada pada data dengan menggunakan berbagai macam fungsi dari transformasi *non-linear* yang ditata secara berlapis-lapis dan mendalam. Teknik dan algoritma dalam *deep learning* dapat digunakan dalam memenuhi kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran takterarah (*unsupervised learning*), dan semi-terarah (*semi-supervised learning*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya. *Deep Learning* juga disebut sebagai *Deep* karena struktur dan jumlah jaringan saraf yang ada pada algoritmanya sangat banyak hingga dapat mencapai hingga ratusan lapisan (Lorentius et al., 2019).

Pendeteksian objek dapat dilakukan salah satunya adalah dengan menggunakan metode *deep learning*. Penggunaan dari *deep learning* didasarkan oleh beberapa faktor, contohnya adalah karena penggunaan data *training* dengan jumlah yang cukup banyak untuk dapat dipelajari dan dianalisis karakteristik dari data *training* tersebut, sehingga komputer memerlukan performa yang cukup tinggi agar data *training* dapat berkembang dengan signifikan (Zhao et al., 2019). Konsep dari algoritma *deep learning* terinspirasi oleh otak manusia (Ertam& Aydin, 2017), sama seperti manusia yang berulang kali belajar dari pengalaman, *deep learning* juga akan melakukan hal yang serupa, yaitu dengan melakukan pengulangan secara terus menerus dalam melakukan tugasnya agar dapat memberikan hasil akhir dengan tingkat akurasi yang tinggi dalam melakukan pendeksteksian suatu objek, melakukan pengenalan suara, menerjemahkan bahasa, dan lain sebagainya (LeCun et al., 2015).

2.3.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu bagian dari metode *deep learning* yang biasanya digunakan untuk melakukan pengolahan data yang berupa gambar atau video. CNN memiliki beberapa *layer*, seperti *input layer*, *output layer*, dan *layer* tersembunyi yang terdiri dari *convolution layer* yang berfungsi untuk mengekstraksi fitur dengan mengatur ukuran dari kernel,

pergeseran setiap kernel, dan menambahkan *padding* pada *input* supaya ukuran dari *output* sama dengan ukuran *input*. *Pooling layer* yang berfungsi untuk dapat mengurangi dimensi agar *layer* berikutnya tidak memerlukan komputasi yang besar. Dan *fully connected layer* yang berfungsi untuk melakukan klasifikasi berdasarkan fitur yang telah diekstrak (Narayan & Muthalagu, 2021). CNN dibuat dengan menggunakan cara *Multi Layer Perceptron* dan banyak digunakan dalam pembuatan aplikasi *computer vision* seperti aplikasi untuk dapat melakukan klasifikasi pada citra, mendekripsi objek, dan melakukan pengenalan pada wajah manusia (Radikto, 2022).

Metode CNN dianggap menghasilkan data yang paling signifikan pada pengenalan citra digital, karena CNN memang diimplementasikan berdasarkan sistem pengenalan pada citra pada *visual cortex* manusia (Bili et al., 2022). Dengan memanfaatkan jaringan dari syaraf tiruan, CNN dapat mempelajari data-data yang telah tersedia dengan bantuan *deep learning*. CNN juga dapat diartikan sebagai sebuah matriks yang memiliki sebuah fungsi untuk dapat melakukan penyaringan pada suatu gambar (Lambacing, et al. 2020). Dengan beberapa lapisan yang digunakan untuk menyaring gambar dari setiap mekanismenya, hal tersebut disebut dengan *training*. Citra pada gambar yang telah disaring akan dipecah menjadi dua kategori, yaitu *sampling* yang merupakan proses pengambilan nilai pecahan atau koordinat ruang, dan *quantitasi* yang merupakan mekanisme dalam melakukan *clustering* dengan nilai tahapan citra kontinu dengan dikategorikan menjadi beberapa *level* (Hermawan, 2021; Putra, 2016).

2.3.2 You Only Look Once (YOLO)

YOLO (*You Only Look Once*) merupakan bentuk algoritma dari *deep learning* untuk dapat melakukan deteksi pada objek secara *real time* dengan memanfaatkan algoritma dari *Convolutional Neural Network* (CNN) agar dapat memproses data yang berupa data visual. YOLO dapat mendekripsi objek dengan menerima sebuah *input* berupa gambar, kemudian akan dibagi menjadi kotak dengan ukuran $S \times S$ untuk membuat *bounding box*, dan sistem akan memprediksi probabilitas kelas berdasarkan dengan tingkat keyakinan sistem dalam mendekripsi sebuah objek (Diwan et al., 2023). Sistem deteksi yang dilakukan oleh YOLO yaitu dengan menggunakan *repurposed classifier* atau *localizer* agar dapat melakukan pendekripsi. Dalam YOLO, citra yang

diterapkan melalui berbagai model kemudian menghasilkan *score* atau nilai yang tinggi akan dianggap sebagai sebuah acuan pendekripsi (Rahma, et al. 2021). Cara kerja dari YOLO adalah membagi citra yang berasal dari gambar atau *frame* video menjadi beberapa *grid* kecil, selanjutnya YOLO akan melakukan prediksi pada kotak pembatas *grid* (*Bounding Box*) dan kelas objek bagi setiap *grid*. Setiap kotak pembatas akan memunculkan skor hasil dari deteksi YOLO yang berdasarkan keyakinan YOLO dalam membaca bahwa objek yang dideteksi benar-benar ada atau tidak, kemudian seluruh kotak pembatas *grid* akan digabungkan dan YOLO akan menyaring skor yang rendah (Gerald & Lubis, n.d.).

YOLO (*You Only Look Once*) digunakan serta dikembangkan untuk dapat meningkatkan dan mempercepat efisiensi dari deteksi objek secara *realtime*, hal ini juga yang menjadi unggulan dari penggunaan YOLO. Dengan menggunakan pendekatan deteksi pada objek berbasis *grid* menggunakan konvolusi jaringan syaraf tunggal (*single-shot*), maka akan memungkinkan YOLO untuk melakukan deteksi pada objek secara langsung pada citra dengan kecepatan yang tinggi. Dalam membagi citra, YOLO akan membaginya dengan menjadikan *grid* sel yang tumpang tindih dan setiap sel akan bertanggung jawab untuk melakukan prediksi pada setiap kotak pembatas objek serta kelas objek yang ada di dalamnya. Sehingga hal tersebut dapat memungkinkan YOLO agar dapat melakukan pendekripsi objek secara paralel serta mempercepat proses deteksi yang terjadi (Nafis Alfarizi et al., 2023).

2.4 Flask dan Penerapan Berbasis CNN dan YOLO

Flask merupakan salah satu framework berbasis Python yang sering digunakan untuk mengembangkan aplikasi web, terutama dalam integrasi model machine learning dan deep learning. Flask dikenal karena sifatnya yang ringan dan modular, yang membuatnya ideal untuk aplikasi yang memerlukan respons cepat dan pengolahan real-time tanpa beban berlebihan dari fitur tambahan yang tidak diperlukan (Grinberg, 2018). Flask mendukung pengembangan REST API, di mana aplikasi web dapat menerima data dari pengguna, mengirimkan data tersebut ke model machine learning seperti CNN dan YOLO, dan memberikan hasil prediksi kembali ke pengguna melalui browser. Dalam konteks penelitian ini, Flask dapat digunakan untuk membangun antarmuka web yang terhubung dengan model deteksi jatuh berbasis CNN dan YOLO. Flask memungkinkan integrasi dengan pustaka TensorFlow atau PyTorch, yang memberikan fleksibilitas dalam memuat

dan menjalankan model deteksi jatuh pada server. Flask juga mendukung pengembangan API endpoints, yang memfasilitasi pengiriman data video dari frontend ke server untuk dianalisis oleh model deteksi jatuh. Studi sebelumnya menunjukkan bahwa Flask telah digunakan secara efektif dalam pengembangan aplikasi berbasis web yang mengintegrasikan deep learning. Misalnya, penelitian oleh Dong et al. (2021) mengembangkan sistem pengenalan wajah real-time berbasis Flask yang terhubung dengan model CNN, menunjukkan bahwa Flask mampu mengakomodasi beban pemrosesan secara efisien. Selain itu, Flask juga dapat diintegrasikan dengan WebSocket untuk streaming data video real-time, memungkinkan hasil deteksi ditampilkan secara instan di frontend (Dong et al., 2021; Grinberg, 2018).

2.5 Dataset

Dalam pengembangan sistem deteksi jatuh, dataset yang digunakan untuk melatih dan menguji model sangat berperan penting dalam menentukan akurasi dan generalisasi model tersebut. Beberapa dataset yang umum digunakan dalam penelitian deteksi jatuh adalah sebagai berikut:

1. Le2i Fall Detection Dataset

Dataset ini terdiri dari video yang merekam berbagai simulasi jatuh dan aktivitas harian normal. Le2i Fall Detection Dataset menyediakan variasi dalam skenario jatuh, seperti jatuh dari duduk, jatuh dari berdiri, dan jatuh ke depan. Dataset ini banyak digunakan dalam pelatihan model deep learning, khususnya CNN, untuk membedakan insiden jatuh dari aktivitas sehari-hari. Le2i juga menyediakan data dalam format yang kompatibel untuk digunakan dalam analisis real-time (Kwolek & Kepski, 2015).

2. UP-Fall Detection Dataset

UP-Fall Detection Dataset merupakan dataset multimodal yang mengandung data dari berbagai sensor dan video untuk mendeteksi insiden jatuh. Dataset ini memiliki berbagai skenario jatuh dan aktivitas yang bervariasi, seperti berjalan, duduk, dan berlari. Dataset ini sangat cocok untuk melatih model CNN dan YOLO karena mencakup variasi postur tubuh dan gerakan yang umum terjadi pada lansia. Dataset ini telah digunakan dalam banyak penelitian sebagai benchmark untuk mengukur akurasi sistem deteksi jatuh (Martínez-Villaseñor et al., 2019).

3. MobiFall Dataset

Dataset ini dikembangkan untuk mendukung penelitian deteksi jatuh yang melibatkan perangkat wearable, seperti ponsel pintar dan akselerometer. Namun, data video yang disediakan juga dapat digunakan untuk melatih model berbasis visi seperti CNN dan YOLO. MobiFall menyediakan data dalam berbagai aktivitas harian yang bisa digunakan sebagai basis untuk mengembangkan model deteksi yang dapat diimplementasikan dalam berbagai perangkat (Vavoulas et al., 2016).

2.6 Statistik Lansia

Organisasi seperti World Health Organization (WHO) melaporkan bahwa sekitar 30% lansia di atas 65 tahun mengalami jatuh setiap tahun, dan persentase ini meningkat menjadi 50% untuk lansia di atas 80 tahun. Statistik ini menunjukkan pentingnya sistem deteksi jatuh yang cepat dan akurat untuk meminimalkan dampak cedera serius seperti patah tulang, trauma kepala, atau bahkan kematian. Semakin tua usia seseorang, semakin tinggi risiko jatuh akibat berkurangnya fungsi fisik, seperti kekuatan otot dan keseimbangan tubuh (WHO, 2021).

2.6.1 Dampak Psikologis pada Lansia Akibat Jatuh

Selain dampak fisik, lansia yang pernah mengalami jatuh sering mengalami ketakutan untuk bergerak atau beraktivitas secara normal, yang dikenal sebagai post-fall syndrome. Ketakutan ini dapat menyebabkan penurunan kualitas hidup dan mempercepat penurunan fisik lebih lanjut, yang berujung pada penurunan kemandirian dan peningkatan ketergantungan pada orang lain atau fasilitas kesehatan (Ambrose et al., 2013).

2.6.2 Risiko Cedera pada Lansia Akibat Jatuh

Jatuh pada lansia dapat menyebabkan cedera serius, seperti patah tulang pinggul, yang memerlukan intervensi bedah dan pemulihan jangka panjang. Menurut data dari *Centers for Disease Control and Prevention* (CDC), lebih dari 95% patah tulang pinggul pada lansia disebabkan oleh jatuh, dan insiden ini sering kali berdampak pada menurunnya kemampuan fisik secara permanen (CDC, 2020). Ini menegaskan pentingnya sistem peringatan dini untuk mendeteksi jatuh dan mengurangi waktu respons dalam memberikan pertolongan.

2.7 Metrik Evaluasi Sistem Deteksi Jatuh

Untuk mengevaluasi performa sistem deteksi jatuh, beberapa metrik kinerja yang umum digunakan meliputi:

1. Akurasi (*Accuracy*)

Akurasi mengukur proporsi total prediksi yang benar dalam keseluruhan dataset. Dalam sistem deteksi jatuh, akurasi mengindikasikan seberapa tepat sistem dalam mengenali insiden jatuh maupun aktivitas non-jatuh. Akurasi yang tinggi menunjukkan bahwa sistem dapat secara konsisten memberikan hasil yang benar, baik dalam mendeteksi jatuh maupun aktivitas harian lainnya.

2. Sensitivitas (*Recall*)

Sensitivitas, atau dikenal juga sebagai *true positive rate*, mengukur kemampuan sistem dalam mendeteksi jatuh dari semua insiden jatuh yang benar-benar terjadi. Sensitivitas tinggi menunjukkan bahwa model mampu mengenali sebagian besar kejadian jatuh dan meminimalkan kesalahan negatif, yaitu insiden jatuh yang tidak terdeteksi (Altman & Bland, 1994).

3. Spesifisitas (*Specificity*)

Spesifisitas mengukur kemampuan sistem dalam mengenali aktivitas normal dan menghindari kesalahan deteksi, atau false positives. Spesifisitas yang tinggi menunjukkan bahwa sistem dapat membedakan dengan baik antara aktivitas normal dan insiden jatuh, yang mengurangi risiko alarm palsu.

4. Presisi (*Precision*)

Presisi mengukur proporsi prediksi positif yang benar-benar jatuh dari total prediksi jatuh. Presisi tinggi berarti model dapat secara akurat mengidentifikasi insiden jatuh tanpa banyak kesalahan deteksi. Metrik ini penting untuk menghindari terlalu banyak alarm palsu yang dapat mengganggu pengguna.

5. F1-Score

F1-Score adalah metrik gabungan dari presisi dan sensitivitas, memberikan gambaran seimbang tentang kinerja model, terutama jika terdapat ketidakseimbangan antara kelas jatuh dan aktivitas non-jatuh. Penggunaan metrik ini dalam penelitian sistem deteksi jatuh memastikan bahwa sistem yang dikembangkan dapat memberikan kinerja yang handal dan relevan

untuk diaplikasikan di lingkungan nyata (Altman & Bland, 1994; Kong et al., 2019).

