

OPTIMASI KINERJA SISTEM PENGENALAN WAJAH *REAL-TIME* BERBASIS CCTV MENGGUNAKAN *MULTITHREADING* PADA PERANGKAT TANPA GPU

William Juniarta Hadiman¹⁾, Yustina Sri Suharini¹⁾

1) Program Studi Teknik Informatika, Institut Teknologi Indonesia

E-mail: williamhadiman@gmail.com

Abstrak

Sistem pengenalan wajah berbasis CCTV secara *real-time* menuntut kinerja komputasi yang tinggi, terutama ketika dijalankan pada perangkat tanpa akselerasi GPU. Keterbatasan sumber daya ini sering menyebabkan penurunan kecepatan pemrosesan serta meningkatnya latensi sistem. Penelitian ini bertujuan untuk mengoptimasi kinerja sistem pengenalan wajah *real-time* berbasis CCTV menggunakan teknik *multithreading* pada perangkat berbasis CPU. Sistem dibangun dengan memanfaatkan MTCNN sebagai metode deteksi wajah, ArcFace sebagai metode ekstraksi fitur, serta Face Image Quality Filtering (FIQA) sebagai penyaring kualitas citra wajah. Fokus utama penelitian ini adalah menganalisis pengaruh *multithreading* terhadap peningkatan *frame per second* (FPS), waktu inferensi, dan penggunaan CPU. Pengujian dilakukan menggunakan lima skenario pengujian yang merepresentasikan berbagai kondisi jumlah objek, pergerakan, dan pencahayaan. Hasil pengujian menunjukkan bahwa penerapan *multithreading* mampu meningkatkan FPS dan menurunkan waktu inferensi secara signifikan dibandingkan dengan sistem *single-thread*, meskipun penggunaan CPU meningkat. Dengan demikian, *multithreading* terbukti efektif dalam mengoptimasi kinerja sistem pengenalan wajah *real-time* pada perangkat tanpa GPU.

Kata Kunci : CCTV, Multithreading, Pengenalan wajah

Pendahuluan

Pemanfaatan sistem pengenalan wajah berbasis CCTV semakin luas digunakan pada berbagai bidang, seperti sistem keamanan, absensi otomatis, serta pemantauan area publik. Kebutuhan akan sistem yang mampu bekerja secara *real-time* menjadi tantangan tersendiri, terutama ketika sistem dijalankan pada perangkat dengan keterbatasan sumber daya komputasi, khususnya tanpa dukungan GPU. Pada kondisi ini, beban pemrosesan untuk deteksi wajah, ekstraksi fitur, dan proses pencocokan identitas dapat menyebabkan terjadinya latensi tinggi serta penurunan *frame rate*.

Salah satu pendekatan untuk mengatasi keterbatasan tersebut adalah dengan menerapkan teknik *multithreading*. *Multithreading* memungkinkan proses pengambilan data (*capture*) dan pemrosesan data (*processing*) berjalan secara paralel, sehingga waktu tunggu antar proses dapat diminimalkan. Selain itu, untuk menjaga kualitas data yang diproses, sistem juga dilengkapi dengan Face Image Quality Filtering (FIQA) untuk menyaring citra wajah yang tidak layak, seperti wajah dengan tingkat blur tinggi, pencahayaan rendah, atau pose ekstrem.

Penelitian ini difokuskan pada analisis pengaruh penerapan *multithreading* terhadap kinerja sistem pengenalan wajah berbasis CCTV pada perangkat tanpa GPU. Parameter kinerja yang dianalisis meliputi *frame per second* (FPS), waktu inferensi, serta penggunaan CPU. Dengan demikian, diharapkan penelitian ini dapat memberikan kontribusi dalam pengembangan sistem pengenalan wajah yang efisien dan dapat diimplementasikan pada perangkat dengan keterbatasan sumber daya.

Studi Pustaka

Pengenalan Wajah (*Face recognition*)

Pengenalan wajah (*Face recognition*) merupakan salah satu aplikasi utama dari *computer vision* yang digunakan untuk mengidentifikasi atau memverifikasi identitas seseorang berdasarkan ciri biometrik pada wajah. Proses ini melibatkan tiga tahap utama yaitu deteksi wajah, ekstraksi fitur, dan pencocokan identitas [1].

Dalam konteks pengawasan CCTV, sistem pengenalan wajah harus bekerja secara *real-time* dan efisien. Tantangan utama muncul dari variasi pencahayaan, pose, serta kualitas video.

Penerapan *pipeline* pengenalan wajah berbasis MTCNN dan ArcFace mampu memberikan akurasi tinggi bahkan pada perangkat non-GPU [1].

Deteksi Wajah Menggunakan MTCNN

Tahap deteksi wajah merupakan fondasi dari sistem pengenalan wajah. Detektor yang baik harus mampu menemukan wajah secara akurat meski dalam kondisi pencahayaan buruk, rotasi wajah, atau adanya penutup sebagian. Salah satu algoritma yang paling banyak digunakan adalah MTCNN (*Multi-Task Cascaded Convolutional Neural Network*).

Arsitektur MTCNN terdiri dari tiga jaringan konvolusional bertingkat:

1. P-Net (*Proposal Network*) untuk menghasilkan kandidat kotak wajah,
2. R-Net (*Refine Network*) untuk menyaring kandidat yang tidak relevan, dan
3. O-Net (*Output Network*) untuk menghasilkan deteksi akhir sekaligus menentukan titik landmark wajah.

Pendekatan bertingkat ini membuat MTCNN efisien dalam mendeteksi wajah berukuran kecil dan besar sekaligus [2]. Kelebihan utama MTCNN adalah kemampuannya beradaptasi pada lingkungan dunia nyata seperti pengawasan CCTV yang dinamis. Algoritma ini mampu mempertahankan akurasi tinggi meski pada citra buram atau gelap [3]. MTCNN dapat dikombinasikan dengan metode ekstraksi fitur seperti ArcFace untuk meningkatkan konsistensi deteksi wajah antar *frame video* [4].

Ekstraksi Fitur Menggunakan ArcFace

Setelah wajah terdeteksi, langkah berikutnya adalah mengekstraksi fitur unik yang membedakan satu individu dengan lainnya. Di sinilah ArcFace berperan penting sebagai metode *feature Embedding* berbasis margin sudut. ArcFace mengoptimalkan jarak antar vektor wajah menggunakan *Additive Angular Margin Loss*, yang memperbesar perbedaan antar identitas sekaligus memperkecil jarak intra-kelas. Pendekatan ini memungkinkan model mengenali individu secara lebih presisi dibandingkan *Softmax loss standar* [5].

Dalam sistem *real-time* tanpa GPU, ArcFace tetap dapat diimplementasikan secara efisien dengan teknik optimasi seperti model *pruning* dan *quantization*. *Embedding* ArcFace tetap stabil dan cepat diproses ketika dikombinasikan dengan *pipeline multithreaded* berbasis CPU [1].

Multithreading dalam Sistem Real-time

Untuk menjaga kinerja tinggi pada perangkat terbatas, konsep *multithreading* menjadi krusial. Teknik ini memungkinkan beberapa tugas dijalankan secara paralel untuk memaksimalkan pemanfaatan CPU dan mengurangi waktu tunggu antar proses. *Multithreading* memungkinkan pembagian beban kerja ke beberapa *thread* dalam satu proses. Dalam sistem pengenalan wajah, ini berarti proses *capture*, *preprocessing*, dan *matching* dapat berjalan bersamaan. Menurut penelitian, pendekatan ini dapat mengurangi *latency* hingga 40% dibanding sistem *single-threaded* [7].

Face Image Quality Filtering (FIQA)

FIQA merupakan komponen tambahan penting dalam sistem pengenalan wajah untuk memastikan hanya citra berkualitas baik yang digunakan dalam proses identifikasi. Hal ini sangat penting dalam sistem CCTV yang sering menghadapi citra buram atau tidak fokus.

FIQA menilai kelayakan citra wajah dengan memperhatikan beberapa aspek seperti ketajaman, pencahayaan, pose, dan penutupan sebagian wajah. Sistem FIQA yang berbasis *relative classifiability* mampu meningkatkan akurasi pengenalan wajah dengan menolak citra berkualitas rendah sebelum proses verifikasi [8]. Selain itu, penelitian lain menunjukkan bahwa penggunaan *confidence-calibrated* FIQA dapat menurunkan *false acceptance rate* hingga 30%, menjadikannya strategi penting dalam pengawasan berbasis CCTV *real-time* [9].

Optimasi Sistem Tanpa GPU

Perangkat pengawasan umumnya tidak dilengkapi GPU karena keterbatasan biaya dan

daya. Oleh karena itu, strategi optimasi pada CPU menjadi aspek kunci dalam sistem pengenalan wajah *real-time*.

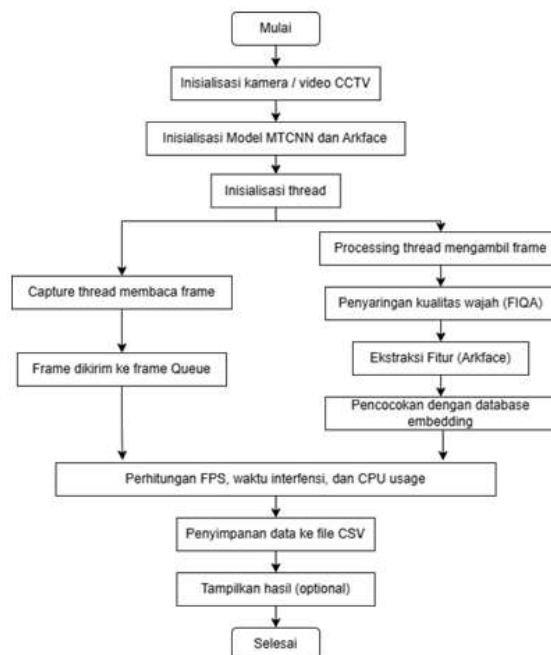
Beberapa pendekatan optimasi umum meliputi:

- *Model Quantization*: Mengubah presisi bobot model menjadi *integer* untuk mempercepat komputasi tanpa kehilangan akurasi signifikan [10].
- *Multithread Scheduling*: Memanfaatkan *pipeline* paralel untuk mengurangi *bottleneck* antar proses [7].
- *Dynamic Thresholding*: Mengadaptasi ambang kualitas FIQA berdasarkan kondisi cahaya dan *noise* [1].

Integrasi *multithreading* dan FIQA terbukti meningkatkan *throughput* sistem hingga 1.8 kali lipat pada platform CPU dibandingkan pendekatan konvensional [1]. Selain itu, penggunaan ArcFace sebagai model utama memperkuat stabilitas pengenalan dengan tetap mempertahankan kecepatan inferensi tinggi.

Alur Penelitian

Sistem yang penulis buat memiliki alur sebagai berikut yang diwakilkan dengan diagram alur sebagai berikut:



Gambar 1. Alur penelitian

Hasil dan Pembahasan

Hasil Pengujian Kinerja Sistem Single-Thread dan Multi-Thread

Tabel 1. Tabel Perbandingan Single-Thread dan Multi-Thread Pada Semua Kasus

Case	Mode	FPS Avg	Infer Avg (ms)	CPU Avg (%)
Case 1 (kosong)	1 Single	24.43	15.78	70.41
Case 1 (kosong)	1 Multi	25.30	19.55	81.11
Case 2 (1 org)	1 Single	8.98	169.11	76.97
Case 2 (1 org)	1 Multi	18.58	142.87	84.11
Case 5 (2 org redup)	2 Single	22.19	47.06	71.10
Case 5 (2 org redup)	2 Multi	24.65	57.44	75.18

Pengujian kinerja dilakukan pada lima skenario pengujian yang merepresentasikan tingkat beban sistem dari ringan hingga berat, yaitu: kondisi tanpa objek, satu orang diam, satu orang bergerak, dua orang dengan pencahayaan cukup, serta dua orang dengan pencahayaan redup. Masing-masing skenario diuji dalam dua mode, yaitu single-thread dan multi-thread. Parameter yang diamati meliputi FPS rata-rata, waktu inferensi rata-rata, dan penggunaan CPU. Berdasarkan hasil pengujian pada Case 1 (tanpa objek), sistem single-thread menghasilkan FPS rata-rata sebesar 24,43 FPS dengan waktu inferensi 15,78 ms, sedangkan mode multi-thread menghasilkan FPS rata-rata 25,30 FPS dengan inferensi 19,55 ms. Pada kondisi ini, perbedaan performa relatif kecil karena tidak terdapat beban deteksi dan pengenalan wajah. Namun, penggunaan CPU pada mode multi-thread meningkat dari 70,41% menjadi 81,11%, yang menunjukkan adanya overhead management thread.

Pada Case 2 (satu orang diam), peningkatan performa akibat multithreading terlihat sangat signifikan. Sistem single-thread hanya mampu mencapai FPS rata-rata sekitar 8,84–8,98 FPS dengan waktu inferensi mencapai 163–169 ms, sehingga sistem tidak lagi memenuhi kriteria real-time. Sebaliknya, sistem multi-thread mampu meningkatkan FPS menjadi 17,48–18,58 FPS dengan waktu inferensi menurun menjadi sekitar 142–144 ms. Hasil ini menunjukkan bahwa multithreading berhasil menggandakan performa sistem pada skenario beban berat berbasis CPU.

Pada Case 3 (satu orang bergerak), sistem single-thread menghasilkan FPS rata-rata 15,83–18,43 FPS dengan waktu inferensi 82–113 ms, sedangkan sistem multi-thread mampu mencapai 26,97–27,23 FPS dengan waktu inferensi yang jauh lebih rendah, yaitu sekitar 43–45 ms. Hal ini menunjukkan bahwa multithreading sangat efektif dalam menangani pergerakan objek yang dinamis. Pada skenario dengan dua orang (Case 4 dan Case 5), sistem multi-thread secara konsisten mempertahankan FPS di atas 24 FPS, sedangkan sistem single-thread cenderung berada di bawah nilai tersebut. Sebagai contoh, pada Case 4_1, FPS sistem single-thread sebesar 20,49 FPS meningkat menjadi 26,47 FPS pada mode multi thread, dengan waktu inferensi turun dari 63,17 ms menjadi 46,18 ms. Hasil ini menegaskan bahwa multithreading mampu menjaga kestabilan sistem pada kondisi multi-objek dan pencahayaan yang lebih kompleks.

Analisis Pengaruh *Multithreading* terhadap FPS dan Waktu Inferensi

Hasil pengujian menunjukkan bahwa penerapan *multithreading* memberikan dampak positif yang sangat signifikan terhadap peningkatan FPS dan penurunan waktu inferensi, khususnya pada skenario dengan beban komputasi tinggi. Pada sistem *single-thread*, seluruh proses akuisisi *frame*, deteksi wajah, ekstraksi fitur, dan pencocokan identitas dilakukan secara berurutan dalam satu alur. Kondisi ini menyebabkan terjadinya *bottleneck* terhadap komputasi, terutama pada tahapan ekstraksi fitur ArcFace.

Pada sistem *multi-thread*, proses pembacaan *frame* dipisahkan dari proses pengolahan citra, sehingga waktu tunggu akibat inferensi dapat dikurangi secara signifikan. Hal ini terbukti dari peningkatan FPS yang mencapai lebih dari 2 kali lipat pada Case 2, serta penurunan waktu inferensi hingga hampir 50% pada Case 3. Dengan demikian, dapat disimpulkan bahwa *multithreading* merupakan strategi optimasi yang efektif untuk menjaga karakteristik *real-time* sistem pengenalan wajah berbasis CCTV pada perangkat tanpa GPU.

Analisis Penggunaan CPU pada Sistem *Multi-Thread*

Walaupun performa FPS meningkat secara signifikan, penggunaan CPU pada sistem *multi-thread* juga mengalami peningkatan. Berdasarkan hasil pengujian, penggunaan CPU rata-rata berada pada kisaran 73–84%, dengan nilai maksimum mencapai 100% pada hampir seluruh skenario. Hal ini menunjukkan bahwa *multithreading* memaksimalkan pemanfaatan seluruh inti prosesor, sehingga seluruh sumber daya komputasi digunakan secara optimal. Namun demikian, kondisi ini masih dapat diterima karena tujuan utama penelitian ini adalah mempertahankan sistem agar tetap berjalan secara *real-time* pada perangkat tanpa GPU. Dengan kata lain, peningkatan beban CPU merupakan konsekuensi langsung dari peningkatan kinerja sistem, dan bukan merupakan indikasi kegagalan optimasi.

Pembahasan Visual Sistem Berdasarkan Hasil Pengujian



Gambar 2. Case 1 (tanpa objek)

Gambar 2 menunjukkan kondisi sistem saat tidak terdapat objek manusia pada area pantauan CCTV. Pada kondisi ini, sistem tetap berjalan pada *FPS* rata-rata sekitar 25 *FPS* dengan penggunaan CPU sebesar 78%, serta tidak menampilkan *bounding box* deteksi. Gambar ini merepresentasikan beban dasar (*baseline load*) sistem saat hanya melakukan proses akuisisi *frame* tanpa deteksi wajah.



Gambar 3. Case 4 (Dua orang dalam pencahayaan normal)

Gambar 2 memperlihatkan sistem berhasil mendeteksi dan mengenali dua wajah secara bersamaan dengan tetap mempertahankan performa *real-time*. Pada kondisi ini, sistem menampilkan *bounding box*, label identitas, nilai kemiripan, serta informasi *FPS*, waktu inferensi, dan penggunaan CPU. Hasil ini membuktikan secara visual bahwa arsitektur *multithreading* mampu menjaga kestabilan pemrosesan meskipun jumlah objek meningkat dan kondisi pencahayaan lebih kompleks.

Diskusi Kelayakan Sistem Tanpa GPU

Berdasarkan seluruh hasil pengujian, dapat disimpulkan bahwa sistem *single-thread* tidak layak digunakan untuk pengenalan wajah *real-time* berbasis CCTV pada perangkat tanpa GPU, terutama pada kondisi terdapat objek manusia. Sebaliknya, sistem *multi-thread* berhasil mengembalikan karakteristik *real-time* sistem dengan *FPS* yang stabil di atas 24 *FPS* pada hampir seluruh skenario pengujian.

Walaupun penggunaan CPU mencapai nilai maksimum pada beberapa kondisi, sistem tetap mampu beroperasi secara stabil tanpa *frame drop* yang signifikan. Dengan demikian, *multithreading* terbukti sebagai pendekatan optimasi yang efektif untuk sistem pengenalan wajah berbasis CCTV pada perangkat CPU-only.

Kesimpulan

Berdasarkan hasil pengujian dan analisis kinerja sistem pengenalan wajah *real-time* berbasis CCTV yang dikembangkan pada perangkat tanpa GPU, dapat disimpulkan bahwa penerapan teknik *multithreading* memberikan peningkatan performa yang signifikan dibandingkan dengan arsitektur *single-thread*. Pemisahan proses antara pengambilan *frame* (*capture thread*) dan pemrosesan deteksi serta pengenalan wajah (*processing thread*) terbukti mampu mengurangi *bottleneck* sistem.

Pada berbagai skenario pengujian, khususnya pada kondisi dengan satu hingga dua objek wajah, *multithreading* mampu meningkatkan nilai *FPS* secara konsisten serta menurunkan waktu inferensi per *frame* secara signifikan. Peningkatan performa paling terasa pada skenario dengan aktivitas objek yang dinamis, di mana beban komputasi meningkat akibat proses deteksi dan ekstraksi fitur yang lebih intensif.

Meskipun terdapat sedikit peningkatan penggunaan CPU pada mode *multithread*, hal tersebut masih berada dalam batas wajar dan tidak menyebabkan degradasi stabilitas sistem. Dengan demikian, *multithreading* terbukti sebagai solusi efektif untuk mengoptimalkan sistem pengenalan wajah berbasis CCTV yang dijalankan pada perangkat dengan keterbatasan sumber daya komputasi, khususnya tanpa dukungan GPU.

Hasil penelitian ini menunjukkan bahwa pendekatan optimasi berbasis *multithreading* dapat dijadikan alternatif yang efisien untuk implementasi sistem pengenalan wajah *real-time* di lingkungan nyata dengan keterbatasan perangkat keras.

Daftar Pustaka

- [1] Imran, A., Ahmed, R., Hasan, M. M., Ahmed, M. H. U., Azad, A. K. M., & Alyami, S. A. (2024). FaceEngine: A Tracking-Based Framework for Real-time Face recognition in Video Surveillance System. *SN Computer Science*, 5(5), 1–15. <https://doi.org/10.1007/s42979-024-02922>
- [2] Jose, E., Greeshma, M., Mithun Haridas, T. P., & Supriya, M. H. (2019). Face recognition based Surveillance System Using FaceNet and MTCNN on Jetson TX2. *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, 608–613. <https://doi.org/10.1109/ICACCS.2019.8728466>
- [3] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503. <https://doi.org/10.1109/LSP.2016.2603342>
- [4] Nguyen, D. D., Le, M. H., Nguyen, X. H., Ngo, H. T., & Nguyen, M. S. (2022). Smart Desk in Hybrid Classroom: Automatic Attendance System based on Face recognition using MTCNN and ARCFACE. *2022 International Conference on Multimedia Analysis and Pattern Recognition, MAPR 2022 - Proceedings*. <https://doi.org/10.1109/MAPR56351.2022.9924775>
- [5] Deng, J., Guo, J., Yang, J., Xue, N., Kotsia, I., & Zafeiriou, S. (2022). ArcFace: Additive Angular Margin Loss for Deep Face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10), 5962–5979. <https://doi.org/10.1109/TPAMI.2021.3087709>
- [6] Baobaid, A., Meribout, M., Tiwari, V. K., & Pena, J. P. (2022). Hardware Accelerators for Real-time Face recognition: A Survey. *IEEE Access*, 10(August), 83723–83739. <https://doi.org/10.1109/ACCESS.2022.3194915>
- [7] Boutros, F., Fang, M., Klemm, M., Fu, B., & Damer, N. (2023). CR-FIQA: Face Image Quality Assessment by Learning Sample Relative classifiability. *Proceedings of the IEEE Computer Society Conference on Computer vision and Pattern Recognition, 2023-June*, 5836–5845. <https://doi.org/10.1109/CVPR52729.2023.00565>
- [8] Ou, F. Z., Li, C., Wang, S., & Kwong, S. (2024). CLIB-FIQA: Face Image Quality Assessment with Confidence Calibration. *Proceedings of the IEEE Computer Society Conference on Computer vision and Pattern Recognition*, 1694–1704. <https://doi.org/10.1109/CVPR52733.2024.00167>
- [9] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *Proceedings of the IEEE Computer Society Conference on Computer vision and Pattern Recognition*, 2704–2713. <https://doi.org/10.1109/CVPR.2018.00286>