

IDENTIFIKASI PERSON PADA GAME FIRST PERSON SHOOTER (FPS) MENGUNAKAN YOLO OBJECT DETECTION DAN DIIMPLEMENTASIKAN SEBAGAI AGENT CERDAS AUTOMATIC TARGET HIT

Rosa Andrie Asmara¹⁾, M. Rahmat Samudra A²⁾, Dimas Wahyu Wibowo³⁾

Politeknik Negeri Malang

Jl. Soekarno Hatta No. 09 Malang, Indonesia

¹⁾*Rosa.andrie@polinema.ac.id*

²⁾*syllxvestern@gmail.com*

³⁾*dimas.w@polinema.ac.id*

Abstrak

Game First-person shooter (FPS) merupakan genre video game yang berpusat pada senjata, umumnya permainan ini membutuhkan akurasi untuk membidik sasaran dengan cepat. Tetapi pemain terutama pemain baru biasanya tidak memiliki reaksi yang cepat dalam mengetahui lawan (person) disekitarnya. Metode yang biasa digunakan adalah memanipulasi memori menggunakan dynamic-link library untuk membuat asisten bidik untuk mendeteksi lawan. Penelitian ini mendeteksi dan mengklasifikasi citra person dengan menggunakan metode YOLO. Versi YOLO yang digunakan sebanyak tiga versi, yaitu YOLOv3, YOLOv4 dan YOLOv5s, karena tingkat keberhasilan YOLO yang luar biasa. YOLO akurasi klasifikasi sebesar 65%, dan presisi sebesar 98% dengan performa inference pada YOLO Tiny sebesar 30FPS. YOLO pada penelitian ini mendapatkan hasil yang baik pada klasifikasi, akan tetapi dibutuhkan metode untuk mempercepat inference yang baik juga untuk melakukan deteksi yang lebih cepat. Metode YOLO dapat digunakan untuk melakukan deteksi objek, akan tetapi pada kecepatan deteksi didapatkan hasil yang kurang memuaskan..

Kata Kunci: Deteksi Person, Asisten Bidik, YOLO.

1. PENDAHULUAN

Game First-person shooter (FPS) adalah game yang bergenre berpusat pada senjata dan pertempuran dalam perspektif orang pertama. FPS sering berfokus pada gameplay aksi, dengan tembak-menembak yang biasanya memberi pemain pilihan senjata, yang memiliki dampak besar pada bagaimana pendekatan atau strategi yang digunakan oleh pemain [1]. Biasanya game FPS bertema militer dengan genre yang memiliki model realistis seperti senjata dengan bentuk aktual yang menggabungkan laju tembakan, ukuran magazine atau jumlah amunisi, dan akurasi [2]. Game First Person Shooting (FPS) adalah game yang cukup populer saat ini. FPS adalah permainan membutuhkan akurasi untuk membidik yang tinggi, dengan menggunakan device mouse pada PC [3]. Tetapi tidak semua pemain memiliki response time yang cepat dalam mengetahui situasi di sekitarnya terutama pada pemain baru yang baru terjun ke dunia game bergenre FPS [4]-[5]. Untuk membantu para pemain terdapat suatu metode yaitu menginjeksi suatu kode program menggunakan dynamic-link library (DLL) untuk memanipulasi memori maupun data asset dari game [6]. Berbeda dengan metode tersebut, penelitian ini menggunakan layar game pemain yang di capture secara real time lalu mendeteksi objek person pada hasil tangkapan layar tanpa perlu menginjeksi kode program ke dalam game Pada penelitian ini penulis

menggunakan metode You Only Look Once (YOLO).

Object Detection adalah teknologi di bidang komputer yang berkaitan dengan Computer Vision dan Image Processing yang dapat mendeteksi suatu benda ataupun manusia dalam gambar maupun video. Kita sebagai manusia dapat dengan mudah dalam mengenali sebuah benda, berbeda dengan komputer. Komputer memerlukan instruksi-instruksi tertentu untuk mengenali sebuah objek. Tujuan dari object detection adalah untuk meniru kecerdasan manusia dalam sebuah komputer. Object detection menentukan keberadaan suatu objek dan ruang lingkungannya serta lokasi pada sebuah gambar, dimana satu kelas mewakili kelas objek dan kelas lain mewakili kelas non-objek. Deteksi objek dapat dibagi lagi menjadi hard detection dan soft detection. Hard detection bisa mendeteksi adanya objek beserta lokasi pada objek tersebut sedangkan Soft detection hanya mendeteksi adanya objek saja [7]-[8].

You Only Look Once (YOLO) adalah model metode Convolutional Neural Network dengan pendekatan baru untuk mendeteksi objek. Pada metode deteksi objek sebelumnya, repurposes classifiers digunakan untuk mendeteksi objek, sebagai gantinya YOLO melakukan frame pada objek yang dideteksi sebagai regression problem untuk membatasi bounding box secara terpisah dengan probabilitas kelas masing-masing bagian. Single Neural Network melakukan prediksi

Bounding Box dan probabilitas dalam satu gambar penuh dalam sekali evaluasi. Kelebihan dari YOLO ini dapat memproses gambar secara real-time pada 45 frame per detik, dan YOLO versi lebih ringan dapat memproses 155 frame per detik, hasil tersebut mencapai dua kali lipat dari deteksi objek realtime lainnya [9]. Berdasarkan permasalahan di atas, maka akan dilakukan penelitian dengan judul "Identifikasi Person Pada Game First Person Shooter Menggunakan YOLO". Penelitian ini bertujuan untuk membantu pemain dalam mendeteksi objek person dalam game FPS, sehingga dapat lebih waspada terhadap situasi disekitarnya.

Deep Neural Networks (DNNs) telah menghasilkan terobosan di sejumlah bidang, termasuk pemrosesan dan pemahaman gambar, pemodelan bahasa, terjemahan bahasa, pemrosesan ucapan, permainan, dan banyak lainnya. Kompleksitas DNN telah meningkat untuk mencapai hasil, yang pada gilirannya telah meningkatkan sumber daya komputasi yang diperlukan untuk melatih neural network. Mixed Precision Training menurunkan sumber daya yang diperlukan dengan menggunakan aritmatika presisi rendah, yang memiliki manfaat mengurangi jumlah memori yang diperlukan dan mempersingkat waktu training atau inferensi, Half precision membagi dua jumlah byte yang diakses, sehingga mengurangi waktu yang dihabiskan di lapisan terbatas memori. GPU NVIDIA menawarkan throughput aritmatika half precision hingga 8x lebih banyak jika dibandingkan dengan single precision. Karena pelatihan DNN secara tradisional mengandalkan format single precision IEEE, fokus dari penelitian ini adalah pada inferensi dengan half precision sambil mempertahankan akurasi jaringan yang dicapai dengan single precision Teknik ini disebut pelatihan mixed precision karena menggunakan representasi presisi tunggal dan setengah[10]-[14].

peneliti tertarik untuk mengembangkan sebuah perangkat lunak atau sistem untuk mendeteksi objek person pada game dan melakukan pembidikan otomatis. Dalam pengembangannya, peneliti menggunakan bahasa pemrograman berbasis Python dengan library OpenCV. Untuk mendeteksi sebuah objek, peneliti menggunakan algoritma YOLO versi 3, 4, dan 5 menggunakan pre-trained dataset COCO.

2. KAJIAN PUSTAKA

Secara khusus, bagian ini menjelaskan metode serangan menggunakan bypass keamanan, pemalsuan dan pergantian memori, dan injeksi DLL, yang merupakan teknik peretasan memori yang representatif. Cara paling umum untuk memblokir injeksi DLL adalah dengan menggunakan anti cheat. Namun, karena anti cheat pada game beroperasi di lingkungan PC yang sama, penyerang dapat melewati penyerang dengan menganalisis anti cheat,

dengan demikian, pertahanan hanya menggunakan anti cheat di PC pengguna terbatas[15].

2.1. Pemalsuan dan perubahan memori

Jika anti cheat dilewati, penyerang kemudian dapat menganalisis game dan mencoba menyerang berdasarkan analisis tersebut. Salah satu metode penyerangan adalah pemalsuan dan perubahan memori. Pemalsuan dan perubahan memori mengubah beberapa opsi dan nilai yang digunakan oleh klien game pada sistem operasi untuk membuat game lebih mudah dan lebih cepat. Pemalsuan dan perubahan memori menggunakan program Cheat Engine untuk memodifikasi memori, dan nilai yang dimodifikasi tercermin dalam game tersebut. Dengan hanya mengubah nilai-nilai pada memori, penyerang dapat mencapai hasil yang diinginkan dalam permainan. Dalam game, metode serangan sederhana ini tidak mungkin, namun, klien game sederhana yang berjalan di PC pengguna rentan terhadap serangan pemalsuan.

2.2. Injeksi DLL

Sebuah "DLL" mengacu pada modul dan program dengan fungsionalitas yang dapat digunakan bersama oleh aplikasi lain [6]. Karena DLL adalah program yang dikembangkan modul, DLL dapat diperbarui dan digunakan kembali dengan lebih mudah daripada program aplikasi. Ketika DLL dimuat ke ruang alamat memori, aplikasi dapat mengaksesnya kapan saja untuk memanfaatkan fungsinya, sehingga mengurangi overhead memori. Di antara DLL termodulasi, ada beberapa DLL penting untuk digunakan di lingkungan Windows. KERNEL32.DLL, USER32.DLL, dan GDI32.DLL adalah beberapa DLL yang harus disertakan dalam proses pengembangan aplikasi. KERNEL32.DLL adalah DLL yang menyediakan fungsi untuk mengontrol memori, proses, dan thread. USER32.DLL adalah DLL yang digunakan untuk mengontrol antarmuka pengguna (UI). Jika ada UI dalam aplikasi, USER32.DLL harus selalu digunakan. Terakhir, GDI32.DLL digunakan untuk menggambar gambar grafik dan menampilkan teks. Mirip dengan USER32.DLL, jika UI, teks, atau gambar digunakan, GDI32.DLL harus disertakan dalam aplikasi. DLL dikategorikan ke dalam fungsi impor dan fungsi ekspor. Fungsi impor hanya digunakan di dalam DLL dan tidak dapat digunakan dalam aplikasi eksternal, sedangkan fungsi ekspor dapat digunakan secara eksternal.

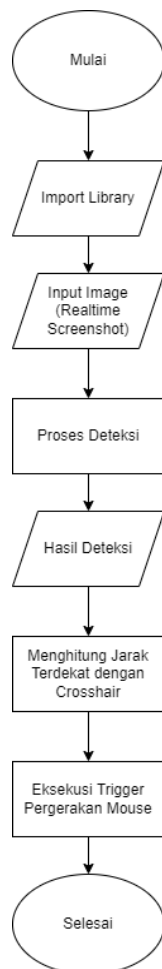
2.3. Load-time dynamic linking

Salah satu proses penting ketika aplikasi berjalan adalah mengimpor daftar DLL untuk digunakan. Karena aplikasi tidak berisi semua kode di DLL, DLL harus ada di jalur yang ditentukan. Jika tidak, aplikasi tidak akan berjalan. Ketika Windows memuat DLL, itu mengalokasikan ruang memori virtual dan memetakan DLL ke memori yang

dialokasikan. Bagian bagian dari program aplikasi menunjukkan informasi DLL yang dipetakan ke memori. Ketika proses pemetaan memori selesai, nama aplikasi dan alamat memori dapat diverifikasi di dalam aplikasi. Selain itu, jika terjadi kesalahan aplikasi, lingkungan dapat digunakan untuk menganalisis sumber kesalahan.

3. METODE

Deteksi objek merupakan sebuah teknik dalam mendeteksi suatu objek atau benda melalui sebuah citra yang dapat mengenali objek tersebut. Citra yang digunakan pada penelitian ini adalah berupa sebuah screenshot yang diambil secara realtime, Alat dan bahan yang digunakan dalam pengembangan ini adalah laptop Asus Zephyrus Duo dengan spesifikasi AMD Ryzen 7 5800H, NVIDIA RTX 3060 dengan RAM 16GB. Dalam pengembangan aplikasi pendeteksi jumlah kendaraan, peneliti menggunakan beberapa tahapan inti yang tersusun dalam sebuah alur aplikasi pada Gambar 1.



Gambar 1. Alur Sistem

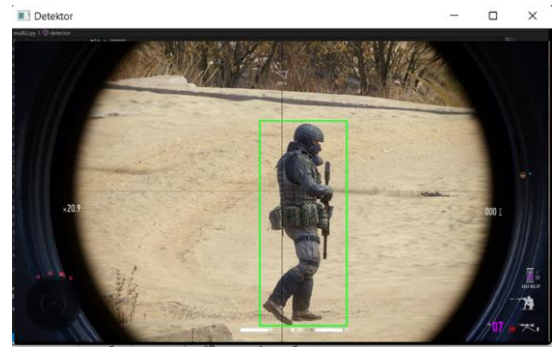
Pada Gambar 1, dapat dilihat bahwa terdapat beberapa tahapan yang diperlukan dalam pengembangan sistem asisten bidik. Tahapan yang digunakan antara lain:

1. Tahap Import Library Pada tahap ini dilakukan sebuah input untuk pengembangan program, yaitu dengan meng-impor library OpenCV sebagai library utama dalam pengembangan aplikasi. Selain penggunaan library Open CV, peneliti juga menggunakan model pre-trained YOLO dengan pemrosesan inferensi menggunakan GPU dan metode inferensi menggunakan Mixed Precision
2. Tahap kedua adalah dengan melakukan input citra digital berupa screenshot yang di capture secara realtime yang akan diproses lebih lanjut oleh model YOLO. input citra akan diresize menjadi 320x320, lalu mengkonversi format input citra menjadi BLOB.
3. Setelah program berjalan dari tahap ketiga, maka proses selanjutnya program akan melakukan deteksi pada citra yang telah dimasukkan. Deteksi dilakukan hanya pada objek person, sehingga objek lain tidak akan ikut terdeteksi. Objek person akan dideteksi menggunakan algoritma YOLO v3, YOLO v4, dan YOLO v5 menggunakan COCO dataset. COCO dataset atau MSCOCO digunakan karena pada sebuah COCO dataset berisi 91 objek dengan total 2,5 juta label dalam 328.000 gambar (Nugroho & Hidayatullah, 2021). Penggunaan YOLO v3, v4, v5 pada tahap deteksi, dilakukan dengan berbagai tahap, yaitu:
 1. Membagi citra dalam region/grid berukuran sxs. Grid-grid tersebut bertanggung jawab untuk mendeteksi sebuah objek. Pada setiap grid dapat diprediksi dengan memberikan sebuah bounding box dengan keterangan nilai confidence. Nilai confidence menunjukkan seberapa akurat deteksi yang dilakukan pada tiap bounding box. Nilai confidence didapat melalui persamaan (3 – 1)
$$Conf(class) = Pr(class) \times IOU \quad (3 - 1)$$
 probabilitas objek yang muncul pada suatu region adalah Pr(Class) dan rasio dari tumpang tindih antara bounding box dan kotak ground truth adalah IOU (Intersection Over Union). Pred adalah luas area pada bounding box, Truth adalah area pada ground truth. Jika nilai IOU semakin besar, maka akan semakin tinggi tingkat akurasi.
 2. Tahap selanjutnya yaitu setiap kotak pembatas memiliki 5 nilai informasi, yaitu x, y, w, h dan c. Nilai x dan y adalah koordinat titik tengah kotak pembatas yang diprediksi, nilai w dan h adalah rasio ukuran lebar dan tinggi relatif terhadap grid, dan c adalah nilai confidence bounding box tersebut.

3. Setelah itu pada setiap grid nilai probabilitas kelas akan diprediksi oleh YOLO jika terdapat objek didalamnya. Saat pengujian, YOLO akan mengkalikan nilai probabilitas kelas dengan nilai confidence dari bounding box. Sehingga menghasilkan nilai confidence class secara spesifik pada tiap bounding box. Bounding box ini kemudian di ekstrak nilai kordinatnya dan kemudian menghitung jarak antara crosshair (titik tengah bidikan). Jarak terdekat akan diseleksi sebagai input untuk diproses.
4. Tahap selanjutnya adalah mengeksekusi trigger mouse movement dengan nilai kordinat yang telah diseleksi sebagai input. Trigger akan membuat simulasi menggerakkan mouse secara otomatis sesuai dengan input yang diberikan.

4. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan menggunakan YOLOv3, YOLOv4, dan YOLOv5s menggunakan masukan file image menggunakan photo viewer secara realtime. Pengujian dilakukan berdasarkan hasil deteksi dan klasifikasi yang diharapkan dengan hasil keluaran oleh sistem. Hasil deteksi yang diharapkan adalah sistem mendeteksi tepat pada objek dan kemudian diklasifikasi sesuai dengan yang objek yang diharapkan. Berikut gambar hasil implementasi.

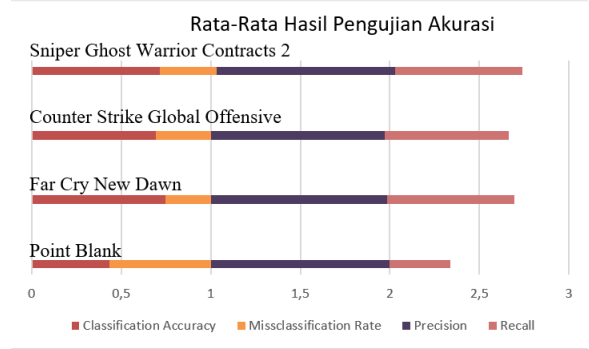


Gambar 4. 1 Hasil implementasi Kondisi Terang



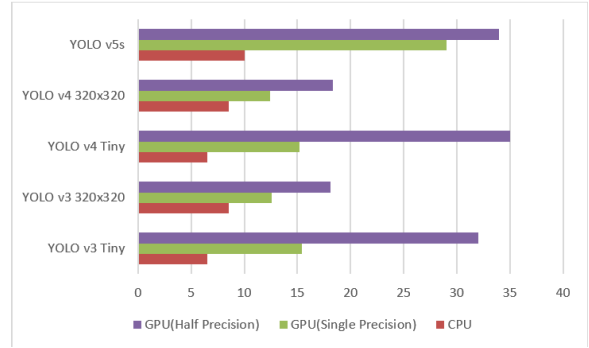
Gambar 4. 2 Hasil implementasi Kondisi Gelap

Berikut diagram perbandingan hasil pengujian yang ditampilkan pada gambar.



Gambar 4.3 Pengujian Akurasi

Dari diagram diatas, dapat diketahui bahwa rata-rata akurasi pada saat klasifikasi cukup baik dengan game far cry new dawn sebesar 75%, Sniper Ghost Warrior Contracts 2 sebesar 72%, Counter Strike Global Offensive sebesar 69% tetapi mengalami penurunan akurasi klasifikasi yang signifikan pada game pointblank sebesar 43% tetapi memiliki presisi yang tinggi yaitu sebesar 100% dengan recall 33%. Untuk misclassification rate yang paling sedikit adalah game Far Cry New Dawn sebesar 25% dengan presisi sebesar 98% dan recall 71%.



Gambar 4.4 Pengujian Performa

Dari diagram diatas, dapat diketahui bahwa rata-rata performa sistem kurang memuaskan. Didapatkan performa single precision paling tinggi ada pada model YOLOv5s sebesar 29FPS, sedangkan paling rendah adalah 12,4FPS oleh model YOLO 320x320 walaupun menggunakan half precision, performa paling tinggi didapatkan oleh model YOLO v4 Tiny sebesar 35 FPS dan terendah ada pada model YOLO v3 320x320 sebesar 18.13 FPS.

5. KESIMPULAN

Pada pengujian akurasi didapatkan nilai rata-rata akurasi deteksi sebesar 43% (Point Blank), 75%(Far Cry New Dawn), 69%(Counter Strike Global Offensive), dan 71%(Sniper Ghost Warrior

Contracts 2) perbedaan hasil tersebut dikarenakan pre-trained model YOLO di training menggunakan cocodataset.org berupa dataset objek person sungguhan bukan model objek person 3D seperti pada game, sehingga menyebabkan penurunan akurasi yang cukup signifikan jika di implementasikan pada game dengan model objek person yang berbeda, selain itu detail grafis pada model objek person 3D juga mempengaruhi hasil deteksi. Berdasarkan hasil penelitian yang telah dilakukan dan pengujian yang didapatkan dapat disimpulkan sebagai berikut:

1. YOLO dapat digunakan untuk merancang sistem yang dapat membantu pemain baru dalam mendeteksi dan membidik objek person. YOLO dapat melakukan deteksi menggunakan metode end-to-end deep learning, akan tetapi pada performa didapatkan hasil yang kurang baik, yaitu pada proses deteksi. Proses inference sangat mempengaruhi performa pengujian, karena proses inference adalah proses untuk mendeteksi. Jika proses inference kurang cepat maka akan mempengaruhi trigger mouse movement yang terjadi delay yang signifikan.
2. Deteksi objek menggunakan YOLO menghasilkan rata-rata akurasi klasifikasi deteksi sebesar 65%, presisi sebesar 98%, dan recall sebesar 61% dengan data uji sejumlah 51 pada masing-masing game.
3. Kecepatan Deteksi tercepat YOLO menghasilkan rata-rata sebesar 33 FPS pada varian YOLO Tiny menggunakan GPU Mixed Precision (Half).

REFERENSI

- [1] Rhee, H. K., Song, D. H., & Kim, J. H. (2019). Comparative analysis of first person shooter games on game modes and weapons – military-themed, overwatch, and player unknowns' battleground. *Indonesian Journal of Electrical Engineering and Computer Science*, 13(1), 116. <https://doi.org/10.11591/ijeecs.v13.i1.pp116-122>
- [2] Song, D. H., Rhee, H. K., & Kim, J. H. (2020). Gender stereotype and hostile sexism among young korean gamers based on teammate selection strategy and game style preferences. *Indonesian Journal of Electrical Engineering and Computer Science*, 19(3), 1512. <https://doi.org/10.11591/ijeecs.v19.i3.pp1512-1518>
- [3] Vicencio-Moreira, R., Mandryk, R. L., Gutwin, C., & Bateman, S. (2014). The effectiveness (or lack thereof) of aim-assist techniques in first-person shooter games. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 937–946. <https://doi.org/10.1145/2556288.2557308>
- [4] Deleuze, J., Christiaens, M., Nuyens, F., & Billieux, J. (2017). Shoot at first sight! First person shooter players display reduced reaction time and compromised inhibitory control in comparison to other video game players. *Computers in Human Behavior*, 72, 570–576. <https://doi.org/10.1016/j.chb.2017.02.027>
- [5] Seya, Y., & Shinoda, H. (2016). Experience and Training of a First Person Shooter (FPS) Game Can Enhance Useful Field of View, Working Memory, and Reaction Time. *International Journal of Affective Engineering*, 15(3), 213–222. <https://doi.org/10.5057/ijae.IJAE-D-15-00014>
- [6] Lee, C. S., Kim, H. K., Won, H. R., & Kim, K. (2021). A method for preventing online games hacking using memory monitoring. *ETRI Journal*, 43(1), 141–151. <https://doi.org/10.4218/etrij.2019-0427>
- [7] Wang, J., Song, L., Li, Z., Sun, H., Sun, J., & Zheng, N. (2021). End-to-End Object Detection with Fully Convolutional Network. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15844–15853. <https://doi.org/10.1109/CVPR46437.2021.01559>
- [8] Venkatesh, A. (2018). Object Tracking in Games using Convolutional Neural Networks [California Polytechnic State University]. <https://doi.org/10.15368/theses.2018.56>
- [9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *ArXiv:1506.02640* [Cs]. <http://arxiv.org/abs/1506.02640>
- [10] Kao, S.-C., Kwon, H., Pellauer, M., Parashar, A., & Krishna, T. (2022). A Formalism of DNN Accelerator Flexibility (arXiv:2206.02987). *arXiv*. <http://arxiv.org/abs/2206.02987>
- [11] Yang, H., Duan, L., Chen, Y., & Li, H. (2021). BSQ: Exploring Bit-Level Sparsity for Mixed-Precision Neural Network Quantization (arXiv:2102.10462). *arXiv*. <http://arxiv.org/abs/2102.10462>
- [12] Cai, Z., & Vasconcelos, N. (2020). Rethinking Differentiable Search for Mixed-Precision Neural Networks (arXiv:2004.05795). *arXiv*. <http://arxiv.org/abs/2004.05795>
- [13] Baharani, M., Mohan, S., & Tabkhi, H. (2019). Real-time Person Re-identification at the Edge: A Mixed Precision Approach (Vol. 11663, pp. 27–39). https://doi.org/10.1007/978-3-030-27272-2_3
- [14] Bruschi, N., Garofalo, A., Conti, F., Tagliavini, G., & Rossi, D. (2020). Enabling Mixed-Precision Quantized Neural Networks in Extreme-Edge Devices. *Proceedings of the 17th ACM International Conference on Computing Frontiers*, 217–220. <https://doi.org/10.1145/3387902.3394038>
- [15] Lee, C. S., Kim, H. K., Won, H. R., & Kim, K. (2021). A method for preventing online games hacking using memory monitoring. *ETRI Journal*, 43(1), 141–151. <https://doi.org/10.4218/etrij.2019-0427>