
Investigasi Pengaruh *Step Training* pada Metode *Single Shot Multibox Detector* untuk *Marker* dalam Teknologi *Augmented Reality*

Vivian Alfionita Sutama¹, Suryo Adhi Wibowo², Rissa Rahmania³

^{1,2,3}Jurusan Teknik Telekomunikasi, Fakultas Teknik Elektro, Telkom University
Jl. Telekomunikasi No.1, Bandung, Indonesia, 40257

^{1,2,3}vivianalfionita@student.telkomuniversity.ac.id, suryoadhiwibowo@telkomuniversity.ac.id,
rissarahmania@telkomuniversity.ac.id

Abstract

Nowadays, Artificial Intelligence is one of the most developing technology, especially on Augmented Reality (AR). AR is a technology which connected between real world and virtual in a real time that allows user to interact directly and display it in 3D. AR technology has two methods, that are AR based on marker and AR based on markerless. However, AR based on marker need an object detection system which has high performance as an interaction tools between user and the device. Single shot multibox detector (SSD) is an object detection algorithm that has fast learning computation and good performance. This method is affected by some parameters like number of epoch, learning rate, batch size, step training, etc. However, to create a good system it took a long process such as taking dataset, labelling process, then training and testing models to gain the best performance. In this experiment, we analyze SSD method in AR technology using inception architecture as pre-trained Convolutional neural network (CNN), and then do transfer learning to minimize amount training time. The configuration that used is the number of step training. The result of this experiment gets the best accuracy in 70.17%. Then, the best performance is used as an object detection model for marker's AR technology.

Keyword: Augmented Reality, Object Detection, Single shot multibox detector, Convolutional neural network, Transfer Learning.

Abstrak

Saat ini, Artificial intelligence merupakan teknologi yang sedang berkembang pesat. Salah satunya adalah teknologi Augmented Reality (AR). AR adalah teknologi yang menggabungkan dunia nyata dengan virtual secara real-time dengan interaksi pengguna secara langsung dan menampilkannya dalam bentuk 3D. Teknologi AR ini memiliki dua metode yaitu dengan marker dan markerless. Dalam perkembangannya, AR berbasis marker membutuhkan sistem deteksi objek yang memiliki performa tinggi sebagai alat interaksi antara pengguna dengan perangkatnya. Single shot multibox detector (SSD) merupakan algoritma deteksi objek yang memiliki komputasi pembelajaran dan kinerja yang baik. Metode ini dipengaruhi oleh beberapa parameter seperti jumlah lapisan konvolusi, epoch, learning rate, jumlah batch, step training, dll. Namun, dalam mengimplementasikannya diperlukan proses yang cukup panjang seperti, pengambilan dataset, proses pelabelan, proses pelatihan menggunakan metode SSD, dan melakukan pengujian terhadap beberapa model untuk mencari performansi paling baik. Dalam percobaan ini, kami melakukan analisis terhadap metode SSD pada teknologi AR menggunakan arsitektur Inception sebagai pre-trained Convolutional neural network (CNN), kemudian dilakukan transfer learning untuk memperkecil jumlah kelas data pelatihan dan waktu pelatihan data. Konfigurasi yang digunakan berupa jumlah step pada pelatihan. Hasil dari penelitian ini menunjukkan akurasi terbaik sebesar 70,17%. Kemudian, performansi terbaik digunakan sebagai model deteksi objek untuk marker pada teknologi AR.

Keyword: Augmented Reality, Deteksi Objek, Single shot multibox detector, Convolutional neural network, Transfer Learning.

I. Pendahuluan

Teknologi informasi semakin berkembang pesat mengikuti perkembangan zaman. Saat ini, teknologi yang memiliki potensi yang besar adalah teknologi kecerdasan buatan. Kecerdasan buatan memungkinkan sistem untuk bekerja seperti manusia dimana sistem dapat belajar berdasarkan pengalaman dan menyesuaikan dengan hal baru. Salah satu teknologi yang berkembang dengan kecerdasan buatan adalah *Augmented Reality* (AR). AR adalah sebuah teknologi yang menggabungkan benda maya baik 2D maupun 3D ke dalam lingkungan nyata lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata [1]. Teknologi AR memiliki dua metode, yaitu AR berbasis *marker* dan AR berbasis *markerless*. Dalam mendukung teknologi AR, diperlukan sistem kecerdasan buatan yang memiliki performa tinggi. Beberapa penelitian telah banyak dilakukan dengan menggunakan metode *machine learning* maupun *deep learning*. Metode yang digunakan antara lain partikel filter [2], *Region-based Convolutional neural network* (RCNN), *Faster RCNN* [3], *You Only Look Once* (YOLO) [4], *Single Shot Multibox Detector* (SSD) [5], dan sebagainya. Hal tersebut bertujuan untuk memberikan hasil deteksi objek, alokasi objek, atau segmentasi pada objek. Selain performansi, efisiensi dan komputasi pada metode yang digunakan juga perlu dipertimbangkan.

Pada penelitian sebelumnya [2], [16], [17], teknologi ini menerapkan metode partikel filter sebagai sistem *tracking*-nya. Partikel filter adalah algoritma deteksi objek yang nilainya bergantung pada representasi sampel atau partikel yang didistribusikan secara acak [16]. Namun, pada hasil penelitian [2], [17] menunjukkan bahwa deteksi *marker* tidak mendeteksi secara maksimal karena masih terdapat banyak noise pada hasilnya. Sedangkan, pada penelitian [16] menunjukkan waktu deteksi objek yang dibutuhkan cukup besar, yaitu 4,98 detik. Semakin berjalannya waktu, teknologi *deep learning* berkembang pesat. Salah satu algoritma *deep learning* yang sedang berkembang adalah algoritma SSD. SSD adalah algoritma *single deep neural network* yang menerapkan fitur *bounding boxes* untuk memperkirakan lokasi objek yang dideteksi [5]. SSD memiliki komputasi dan nilai kecepatan deteksi yang tinggi sehingga metode ini cocok untuk deteksi secara *real-time*. Pada penelitian ini, kami menganalisis performansi algoritma SSD pada teknologi AR dengan arsitektur jaringan *pre-trained Inception V2* menggunakan *dataset* berupa jari tangan. Parameter performansi yang digunakan berupa jumlah *step* pelatihan data. Setelah dilakukan uji coba, maka hasil yang memiliki performansi terbaik digunakan untuk model deteksi jari tangan pada sistem teknologi AR.

II. Metodologi Penelitian

2.1 *Augmented Reality*

Augmented Reality (AR) adalah suatu teknologi yang menggabungkan sebuah konten nyata dan virtual secara *real-time* yang di proyeksikan pada dunia nyata untuk mempermudah penggunaanya dalam melakukan interaksi dengan konten dua dimensi maupun 3 dimensi secara langsung [1]. Teknis sistem AR adalah sebagai berikut:

- Teknologi yang digunakan memiliki tampilan yang dapat menggabungkan antara konten virtual dan nyata,
- Sistem komputer dapat menghasilkan grafik interaktif yang dapat memproses input pengguna secara *real-time* dan memiliki sistem *tracking* yang dapat menemukan posisi sudut pandang pengguna,
- Sistem dapat mengeluarkan data virtual ke dunia nyata.

Teknologi AR dibagi menjadi dua jenis, yaitu AR berbasis *marker* dan AR berbasis *markerless*. Setiap jenisnya memiliki perbedaan masing-masing yang digunakan sesuai dengan kebutuhan. Perbedaan dari keduanya adalah sebagai berikut:

- AR berbasis *marker* adalah AR yang bekerja dengan menggunakan *marker* sebagai *pointer* untuk interaksinya. *Marker* merupakan ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. *Marker* digunakan agar perangkat dapat mengenali posisi dan orientasi objek dan menciptakan benda virtual 3D di koordinat yang ditunjukkan *marker*.

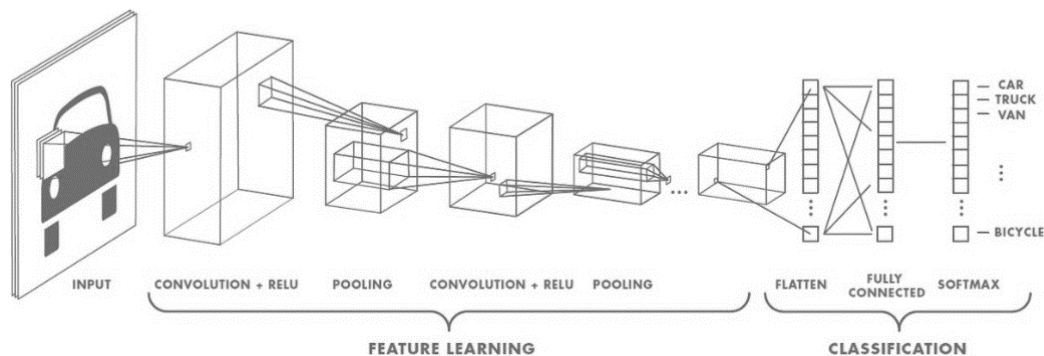
- AR berbasis *markerless* tidak memerlukan *marker* sebagai *pointer* dalam berinteraksi. AR *markerless* bekerja dengan melakukan pemindaian terhadap objek nyata, dengan ruang lingkup yang lebih luas dibandingkan *marker* AR. Contoh *markerless* yang berkembang adalah AR dengan menggunakan *location based* atau GPS.

2.2 Deteksi Objek

Deteksi objek adalah salah satu cabang dari teknologi computer vision yang digunakan untuk mendeteksi suatu objek dari kelas tertentu pada suatu citra visual baik berupa gambar, video, maupun real-time melalui kamera. Sistem deteksi objek bekerja melalui tahapan *preprocessing*, ekstraksi fitur, dan klasifikasi. *Preprocessing* adalah pengolahan data asli untuk meningkatkan kualitas dari data sebelum diolah ke tahap berikutnya. Ekstraksi dan klasifikasi adalah sebuah proses untuk mendapatkan fitur dari data yang digunakan agar objek dapat dikategorikan jenisnya. Salah satu cara pada proses ekstraksi dan klasifikasi adalah menggunakan *deep learning* yaitu *Convolutional neural network* (CNN) [6]. Metode-metode yang dapat digunakan untuk deteksi objek antara lain *Region-based Convolutional neural network* (RCNN), *Faster RCNN* [3], *You Only Look Once* (YOLO) [4], *Single shot multibox detector* (SSD) [5], dan sebagainya. Deteksi objek dapat diimplementasikan pada berbagai aplikasi di kehidupan, misalnya *video surveillance*, *face recognition*, maupun *visual tracking* [7], [8], [9].

2.3 Convolutional neural network

Convolutional neural network (CNN) adalah algoritma *deep learning* yang menggunakan metode pengembangan *Multi Layer Perceptron* (MLP) untuk data dimensi tinggi (gambar atau video). CNN berfungsi untuk klasifikasi citra, segmentasi semantik, deteksi objek, dan ekstraksi ciri [10]. Cara kerja dari CNN yaitu dengan meniru kerja otak pada manusia yang terdiri dari berbagai *neuron*.



Gambar 1. Arsitektur *Convolutional neural network* [11].

Gambar 1 menunjukkan bahwa arsitektur CNN memiliki beberapa lapisan tersembunyi (*hidden layer*) dimana masukan dari CNN adalah berupa vektor tunggal dan data yang dipropagasikan adalah data dua dimensi sehingga operasi linier pada CNN menggunakan konvolusi. Arsitektur CNN dibagi menjadi dua fitur, yaitu *feature extraction* dan *classification*. *Feature extraction* terdiri dari *convolutional layer*, *REL-U*, dan *pooling layer*. Sedangkan, *classification* terdiri dari *fully connected layer*.

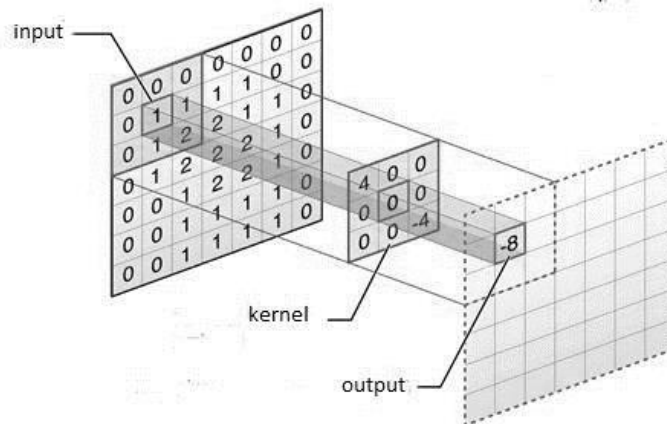
2.3.1 Convolutional Layer

Convolutional layer adalah *layer* yang terdiri dari *neuron* yang tersusun sehingga membentuk sebuah *filter* yang berfungsi untuk deteksi dan mengekstraksi ciri pada *input* yang diberikan [12]. Cara kerja dari *convolution layer* ini adalah melakukan konvolusi blok citra dengan matriks filter. Kemudian, hasilnya akan berupa angka yang merupakan *dot product* antara bagian *input* dengan *filter* yang digunakan. Setelah itu, dengan menggunakan konvolusi *filter* pada kemungkinan posisi *filter* tersebut akan menghasilkan sebuah

activation map. Pada *convolutional layer* terdapat *learnable parameters* (y) yang akan dilatih, persamaan *learnable parameters* dapat dirumuskan sebagai:

$$y = (K * Ch + b) * D, \tag{1}$$

dengan K , Ch , b dan D merepresentasikan ukuran filter / kernel, jumlah kanal, jumlah bias, dan depth. Proses kerja pada *Convolutional Layer* dapat dilihat pada Gambar 2.



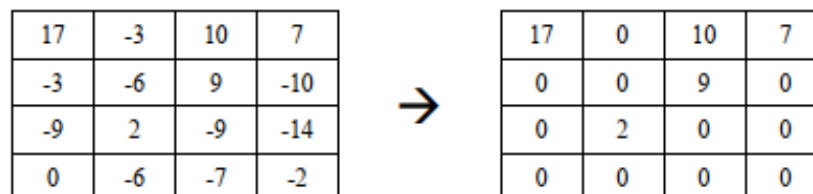
Gambar 2. Proses pada *Convolutional Layer* [12].

2.3.2 Rectified Linier Unit (REL-U)

Rectified Linier Unit (REL-U) adalah fungsi aktivasi pada jaringan *neural network*. Fungsi aktivasi ini terdapat fungsi *non-linier* yang menentukan nilai keluaran ketika mendapat masukan tertentu. Nilai keluaran yang didapat akan bernilai 0 jika negatif dan bernilai sama dengan nilai keluarannya apabila positif. Secara matematis dapat dituliskan sebagai berikut:

$$REL - U(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0, \end{cases} \tag{2}$$

dimana variabel x adalah hasil keluaran dari fungsi aktivasi REL-U. Contoh perhitungan pada REL-U dapat dilihat pada Gambar 3.

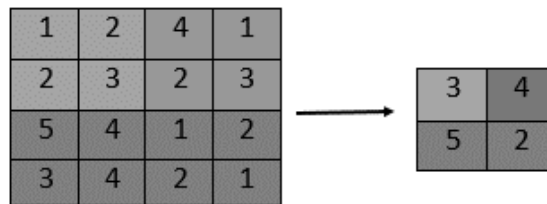


Gambar 3. Proses Perhitungan REL-U.

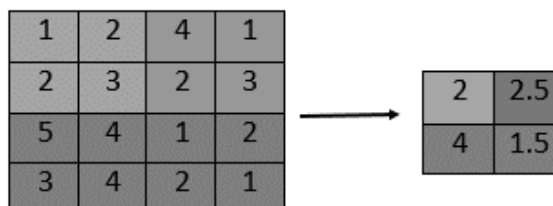
2.3.3 Pooling layer

Pooling layer adalah lapisan yang mereduksi dimensi dari *feature map* secara parsial menggunakan operasi *down-sampling* sehingga dapat mempercepat komputasi dan mengatasi *overfitting*. Fungsi *pooling* yang digunakan ada 2, yaitu *max pooling* dan *average pooling* [12]. Konsep *max pooling* ini adalah dengan menentukan nilai maksimum pada tiap pergeseran filter dan mengambil pixel terbesar untuk menjadi matriks

baru. Sedangkan *average pooling* mereduksi dengan menentukan rata-rata pada tiap pergeseran filternya. Proses matematis pada *pooling layer* dapat dilihat pada Gambar 4, 5.



Gambar 4. Proses *Max Pooling* menggunakan stride 2.



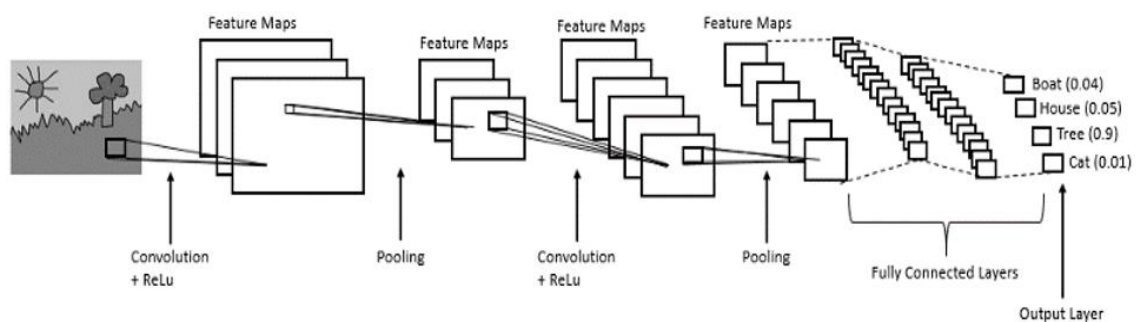
Gambar 5. Proses *Average Pooling* menggunakan stride 2.

2.2.4 Fully-Connected Layer

Fully-connected layer adalah lapisan yang berfungsi dalam penerapan *multi layer perceptron* untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linier [13]. Cara kerja *layer* ini sama seperti jaringan saraf manusia. Lapisan ini beroperasi pada *layer* yang disebut *hidden state* atau *hidden layer*. Secara matematis, keluaran dari *layer* ini dapat dirumuskan sebagai:

$$h(x) = g(b + \sum_i W_i X_i), \tag{3}$$

dimana W_i adalah nilai bobot pada *hidden layer*, X_i adalah nilai masukan, b sebagai nilai bias, dan g adalah fungsi aktivasi. Arsitektur dari *layer* ini dapat dilihat pada Gambar 6.

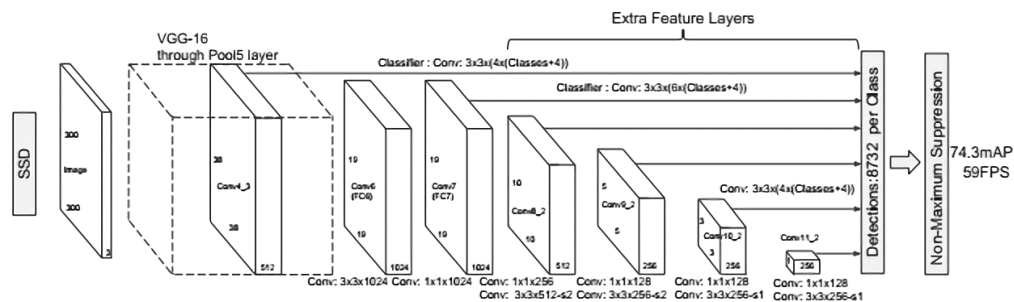


Gambar 6. Arsitektur Fully Connected *Layer* [12].

2.3 Single Shot Multibox Detector

Single shot multibox detector (SSD) adalah algoritma *deep learning* yang bekerja berdasarkan pendekatan pada *feed-forward convolutional network* yang mendiskritisasi ruang output dari kotak pembatas menjadi satu set kotak standar pada berbagai rasio dan skala aspek per lokasi peta fitur [14]. SSD merupakan

algoritma *single deep neural network* yang menerapkan fitur *bounding boxes* untuk memperkirakan lokasi objek yang dideteksi [5].



Gambar 7. Arsitektur SSD [14].

Pada Gambar 7 dapat dilihat bahwa SSD memiliki dua arsitektur jaringan yaitu jaringan VGG-16 dan jaringan pada *extra feature layers*. Jaringan VGG-16 digunakan sebagai *base-network* karena VGG-16 memiliki kinerja yang kuat untuk citra dengan kualitas yang tinggi. Selain itu, SSD menambahkan lapisan fitur konvolusional ke ujung *base network* yang berfungsi memprediksi rasio aspek yang berbeda. Arsitektur jaringan yang dapat digunakan pada *extra feature layer* dapat berupa arsitektur *Inception*, *Mobilenet*, ataupun *Resnet*. SSD memiliki 3 fitur utama [14], yaitu:

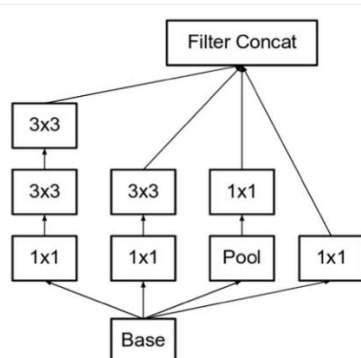
- *Multi-Scale Feature maps*
- *Convolutional Predictor*
- *Default Box and Aspect Ratio*

Dalam SSD terdapat 2 tipe parameter yaitu model parameter dan *hyperparameter*. Parameter adalah konstanta model yang sudah ditentukan pada set data pelatihan dan tidak dapat diubah. Sedangkan, *hyperparameter* adalah konstanta yang dapat disesuaikan dengan kebutuhan untuk mendapatkan model dengan kinerja yang optimal. Macam-macam *hyperparameter* yaitu:

- *Learning Rate*
- *Number of Step training*
- *Number of Epoch*
- *Etc.*

2.4 Arsitektur Inception V2

Inception network adalah suatu arsitektur jaringan pada *neural network* yang menggunakan fitur faktorisasi *layer*. Fitur faktorisasi *layer* ini digunakan untuk mengurangi kedalaman pada jaringan tersebut, sehingga dapat mengurangi representasional bottleneck [15] dimana *neural network* dapat berfungsi lebih baik apabila konvolusi tidak mengubah dimensi input pada citra secara drastis. Selain itu, penggunaan fitur faktorisasi juga membuat proses perhitungan kompleks pada *neural network* menjadi lebih efisien. Model konfigurasi Inception V2 dapat dilihat pada Gambar 8.



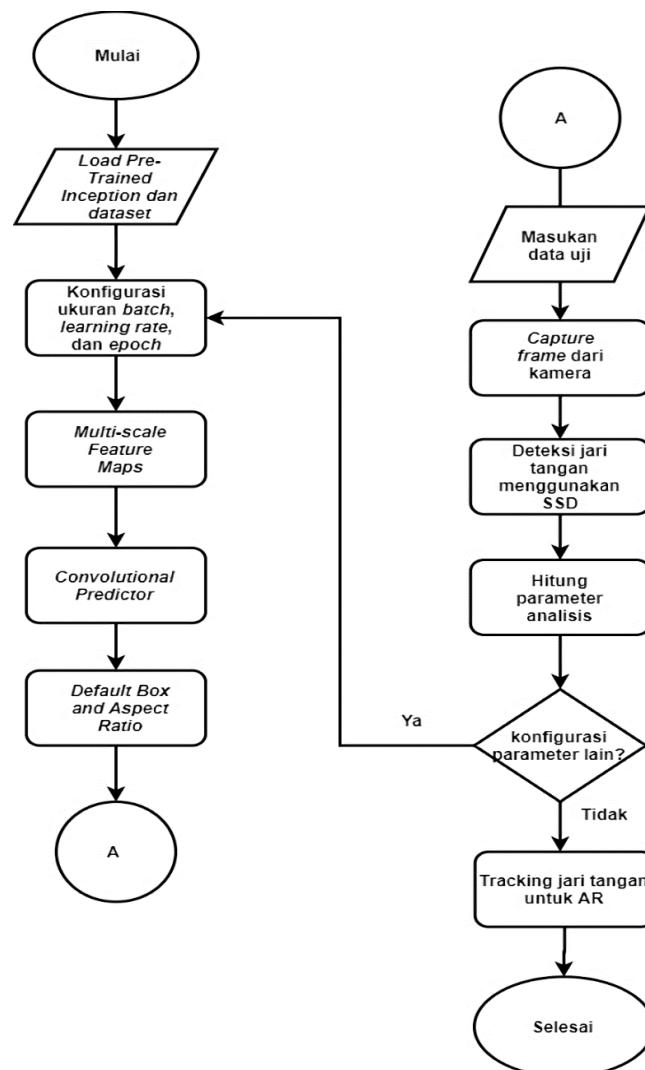
Gambar 8. Model Jaringan Inception V2 [13].

2.4 Transfer Learning

Transfer learning adalah metode dimana network mempelajari hal baru dengan menggunakan kembali pengetahuan yang sudah dilatih pada network sebelumnya. Salah satu cara *transfer learning* adalah menggunakan *fine tuning*. *Fine tuning* yaitu memberikan *dataset* baru dan menyesuainya dengan pengetahuan atau bobot dalam sistem tersebut untuk mengklasifikasikan objek lain yang berkaitan dengan kelas yang sudah ada.

2.5 Perancangan Sistem

Penelitian ini menggunakan algoritma SSD untuk proses deteksi objek. SSD digunakan untuk melakukan ekstraksi ciri dan melakukan klasifikasi citra. Arsitektur yang digunakan yaitu arsitektur jaringan *Inception V2* sebagai pre-trained CNN, kemudian dilakukan *transfer learning* dari *dataset* COCO sebanyak 80 kelas dan mengubahnya menjadi 1 kelas klasifikasi, yaitu jari tangan. Alur kerja sistem dapat dilihat pada Gambar 9.



Gambar 9. Alur kerja sitem penelitian.

Penelitian ini menggunakan 3 parameter performansi yaitu akurasi, *Intersect Over Union* (IoU), dan presisi. Parameter-parameter tersebut berfungsi untuk menentukan model yang terbaik untuk digunakan sebagai *marker* pada teknologi AR. Persamaan matematis parameter tersebut dijabarkan sebagai berikut:

$$IoU = \frac{Bgt \cap Bac}{Bgt \cup Bac}, \quad (4)$$

di mana Bgt adalah *bounding box groundtruth* dan Bac adalah *bounding box actual* yang terdeteksi.

Parameter IoU digunakan untuk menghitung *success plot* pada sistem deteksi. Perhitungan IoU dilakukan berdasarkan area *overlap* atau luas area yang berpotongan antara *actual bounding box* dengan *groundtruth*. Deteksi dikatakan berhasil apabila nilai $IoU \geq 0.5$. Maka, untuk menghitung nilai akurasi dapat menggunakan persamaan berikut:

$$Akurasi = \frac{\sum \text{data benar}}{\sum \text{seluruh data}} \times 100\%, \quad (5)$$

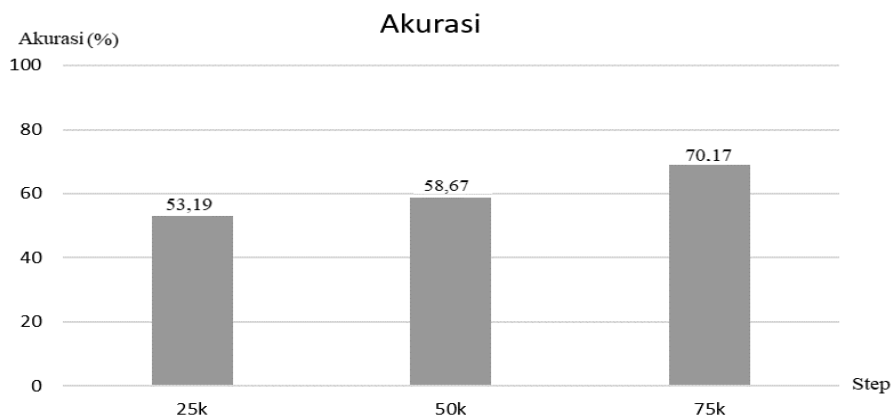
Sedangkan, parameter presisi berfungsi untuk mencari jarak antara *centroid groundtruth* dengan *centroid bounding box actual*. Parameter presisi dikatakan baik apabila nilainya mendekati 0. Persamaan matematis parameter presisi dapat dijabarkan sebagai berikut:

$$E = \sqrt{(Cxgt - Cxac)^2 + (Cyg - Cyac)^2}, \quad (6)$$

Persamaan (6) menunjukkan bahwa E merupakan parameter presisi, Cxgt merupakan *centroid ground truth* untuk koordinat x, Cxac merupakan *centroid actual* untuk koordinat x, Cygt merupakan *centroid ground truth* untuk koordinat y, dan Cyac merupakan *centroid actual* untuk koordinat y. parameter-parameter yang sudah dihitung berfungsi untuk menentukan nilai performansi sistem sehingga parameter tersebut dapat dijadikan acuan sebagai langkah awal untuk optimasi agar deteksi objek untuk *marker* pada AR dapat berjalan mulus dan akurat.

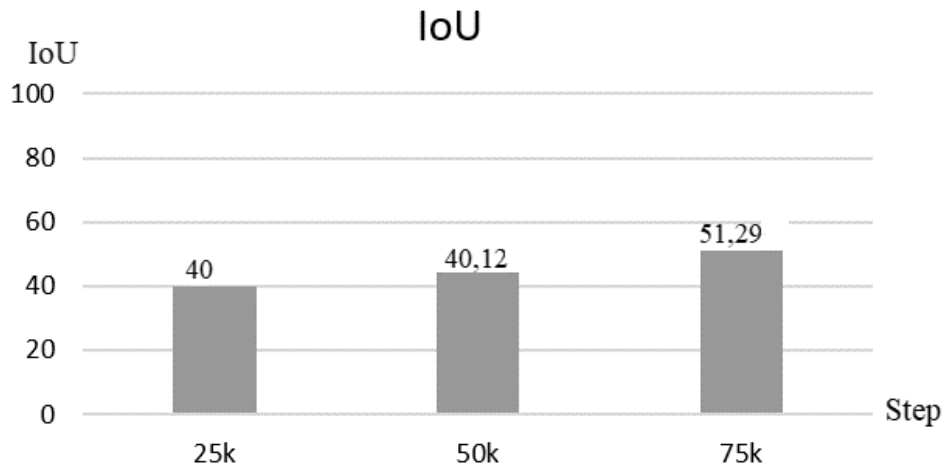
III. Hasil dan Pembahasan

Pada penelitian ini dilakukan 3 kali percobaan dengan jenis konfigurasi berupa jumlah *step* pelatihan sebesar 25000, 50000, dan 75000. Dari 3 percobaan tersebut, kami menghitung akurasi tiap model, *Intersect Over Union* (IoU), dan presisi yang didapatkan. Dari perhitungan persamaan (4), (5), dan (6) didapatkan hasil sebagai berikut:



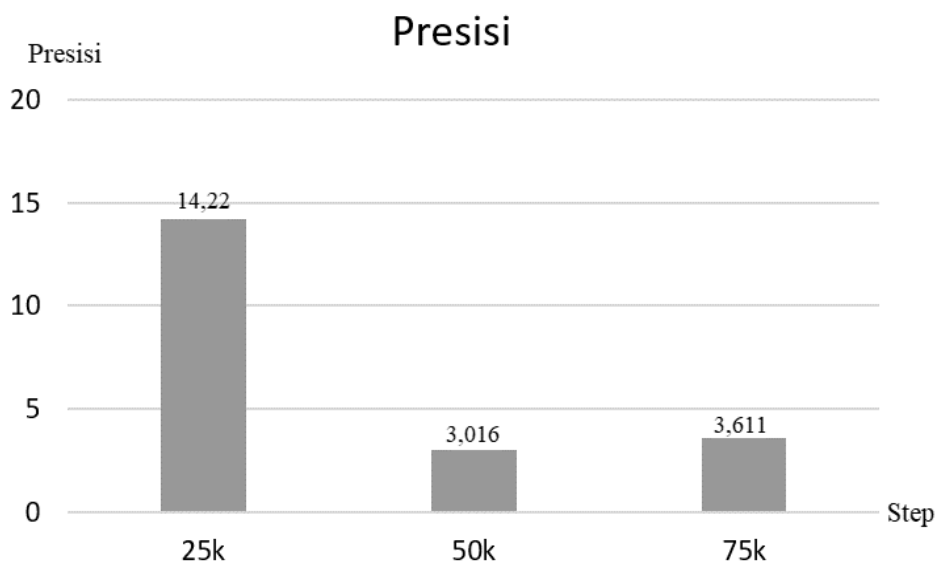
Gambar 9. Hasil Akurasi pada Deteksi Objek

Gambar 9 menunjukkan bahwa parameter akurasi semakin meningkat ketika *step training* semakin bertambah. Hal tersebut dikarenakan semakin banyak *step training* yang digunakan maka validasi sistem akan terus diperbarui sehingga *loss* pada setiap *step*-nya akan semakin kecil nilainya. Parameter akurasi ditentukan berdasarkan perhitungan IoU. Dimana, akurasi meningkat apabila $\text{IoU} \geq 0.5$.



Gambar 10. Hasil Average IoU pada Deteksi Objek.

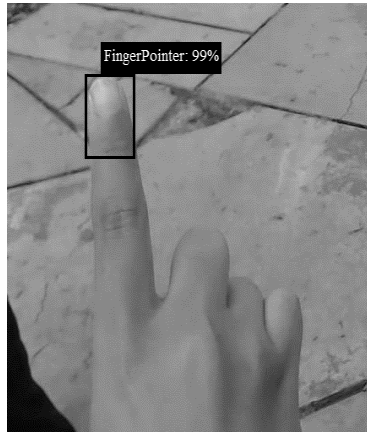
Gambar 10 menunjukkan perbandingan IoU pada masing-masing *step training*. Berdasarkan grafik, semakin besar jumlah *step training* maka hasil peromansi IouU cenderung meningkat. Jika nilai IoU mendekati 1 maka peromansi sistem akan meningkat. Hasil IoU tertinggi terdapat pada *step* ke 75000 dengan nilai sebesar 51,29%.



Gambar 11. Hasil Presisi pada Kondisi *Outdoor*.

Gambar 11 menunjukkan hasil presisi dari setiap percobaan. Dari grafik tersebut, dapat dilihat bahwa semakin banyak *step training* maka nilai presisi sistem cenderung mengecil karena dengan banyaknya *step*

training maka sistem akan semakin banyak belajar dan dapat mengetahui secara detail pada datanya. Sehingga, jarak antara *actual bounding box* yang terdeteksi dengan *groundtruth* yang telah dilabel tidak terlalu jauh. Semakin mendekati 0 nilai presisi maka semakin bagus performansinya. Hal tersebut dikarenakan, semakin kecil nilai presisi maka pergerakan *pointer* pada sistem AR akan lebih cepat / lebih interaktif.



Gambar 12. Pengaplikasian Model Terbaik Deteksi Objek untuk *marker* AR.

Tingkat akurasi terbesar pada pengujian menunjukkan nilai sebesar 70,17% pada *step training* ke 7500. Maka dari itu, model pada *step training* ke 7500 digunakan sebagai model deteksi objek untuk *marker* pada AR. Contoh pengaplikasian deteksi objek untuk *marker* pada AR dapat dilihat pada Gambar 12.

IV. Kesimpulan

Pada penelitian ini, dapat disimpulkan bahwa jumlah *step training* mempengaruhi parameter performansi pada sistem deteksi objek. Hasil terbaik didapat pada *step training* ke 75000 dengan nilai akurasi sebesar 70,17%, IoU sebesar 51,29%, dan presisi sebesar 3,61. Performansi sistem akan semakin meningkat apabila jumlah *step training* bertambah karena semakin banyak jumlah *step training*, maka semakin banyak sistem belajar tentang data yang ada didalamnya. Sehingga, sistem akan memperbarui parameternya secara berkala dan menyebabkan *loss* pada sistem berkurang. Akurasi yang tinggi dan nilai presisi yang rendah dapat meningkatkan performansi deteksi objek sebagai *marker* pada teknologi AR. Hal tersebut dikarenakan tingginya akurasi memberikan deteksi secara akurat. Sedangkan, nilai presisi yang rendah yaitu mendekati 0 akan memberikan deteksi pada gerakan *pointer* secara mulus dan posisi yang tepat. Selain itu, SSD memiliki kecepatan deteksi yang tinggi sehingga cocok untuk deteksi secara *real-time*.

V. Daftar Pustaka

- [1] Billinghamurst, M., Clark, A., & Lee, G. 2015. *A Survey of Augmented Reality*. Foundations and Trends in Human-Computer Interaction, 8(2-3), 73-272.
- [2] S. Sadhana Rao. 2010. *Sixth sense technology*. International Conference on Communication and Computational Intelligence (INCOCCI), Erode.
- [3] S. Ren, K. He, R. Girshick, and J. Sun. 2017. *Faster r-cnn: towards real-time object detection with region proposal networks*. IEEE Transactions on Pattern Analysis & Machine Intelligence.
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. 2016. *You only look once: Unified, real-time object detection*. Proceedings of the IEEE conference on computer vision and pattern recognition.
- [5] M. A. Afwani, E. Utami, and E. Pramono. 2017. *Modifikasi Default-Boxes Pada Model SSD Untuk Meningkatkan Keakuratan Deteksi*. Jurnal IT CIDA.
- [6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE.

- [7] S. A. Wibowo, H. Lee, E. K. Kim, and S. Kim. 2018. *Collaborative learning based on convolutional features and correlation filter for visual tracking*. International Journal of Control, Automation and Systems.
- [8] S. A. Wibowo, H. Lee, E. K. Kim, and S. Kim. 2017. *Convolutional shallow features for performance improvement of histogram of oriented gradients in visual object tracking*. Mathematical Problems in Engineering.
- [9] S. A. Wibowo, H. Lee, E. K. Kim, and S. Kim. 2017. *Visual Tracking Based on Complementary Learners with Distractor Handling*. Mathematical Problems in Engineering.
- [10] J. Wu. 2017. *Introduction to convolutional neural networks*. National Key Lab for Novel Software Technology. Nanjing University. China.
- [11] N. Sofia, 2018. *Convolutional neural network*. A Medium Corporation.
- [12] R. Darmadi. 2018. *Mengenal Convolutional Layer Dan Pooling layer*. Medium Corporation.
- [13] A. Yanuar. 2018. *Fully-Connected Layer CNN dan Implementasinya*. Universitas Gadjah Mada Menara *Machine learning*.
- [14] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. 2016. *SSD: Single shot multibox detector*. Lecture Notes in Computer Science, 21–37.
- [15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. *Rethinking the inception architecture for computer vision*. Proceedings of the IEEE conference on computer vision and pattern recognition.
- [16] V. Rahmawan, D. Oktavian, and D. Alamsyah. 2017. *Penerapan Algoritma Particle Filter pada Face Tracking*.
- [17] Tian, P. 2015. *A particle filter object tracking based on feature and location fusion*. 6th IEEE International Conference on Software Engineering and Service Science (ICSESS).