

Simulasi Penerjemah SIBI (Sistem Isyarat Bahasa Indonesia) Menggunakan Tensorflow Dan Convolutional Neural Network (CNN)

Simulation Of SIBI Translator (Sistem Isyarat Bahasa Indonesia) Using Tensorflow And Convolutional Neural Network (CNN)

1st Muhammad Azka
Imaddudin

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
azkaudin@student.telkomuni-
versity.ac.id

2nd Ida Wahidah Hamzah
Fakultas Teknik Elektro

Universitas Telkom
Bandung, Indonesia
wahidah@telkomuniversity.a-
c.id

3rd Sri Astuti

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
sriastuti@telkomuniversity.ac-
.id

Abstrak—Bahasa Isyarat biasanya digunakan oleh penyandang disabilitas tunarungu untuk berkomunikasi dengan orang lain. Permasalahan yang sering terjadi dalam komunikasi menggunakan bahasa isyarat adalah ketika orang belum mengerti berkomunikasi dengan penyandang disabilitas tunarungu. Penelitian ini membuat simulasi penerjemah SIBI (Sistem Isyarat Bahasa Indonesia) secara *realtime* dan *non realtime*. Dataset (kumpulan gambar bahasa isyarat) total 1200 gambar untuk 6 kelas huruf (A, B, C, D, E, dan F). Kemudian dataset dipisahkan untuk pelatihan 90% dan evaluasi 10%. Sebelum pelatihan dengan model, dataset dianotasikan perkelasnya. Model pelatihan yang digunakan yaitu SSD MobileNet V2. Dataset dilatih dengan 3 skema yaitu 10.000 step untuk skema 1. 20.000 step untuk skema 2. 30.000 step untuk skema 3. Setelah dilatih model akan dievaluasi untuk membangun model prediksi deteksi. Terakhir membuat program deteksi secara *realtime* dan *non realtime*. Dalam penelitian ini hasil yang dicapai bahwa simulasi berkerja dengan baik. Rata-rata tingkat klasifikasi *realtime* 91% dan tingkat klasifikasi *non realtime* 60%.

Berdasarkan hasil pelatihan dan evaluasi untuk semua skema menghasilkan rata-rata learning rate 0,068 iteration/steps, rata-rata total loss 0,32 , rata-rata loss localization 0,067, rata-rata loss classification 0,13, mAP 0,75, dan mAR 0,78. Berdasarkan hasil tersebut sudah cukup baik dalam menerjemahkan Bahasa isyarat.

Kata kunci — SIBI, Deteksi Objek, Tuna Rungu, Tensorflow, SSD MobileNet V2, CNN

I. PENDAHULUAN

Permasalahan yang sering terjadi dalam komunikasi menggunakan bahasa isyarat adalah ketika orang belum terbiasa berkomunikasi dengan penyandang disabilitas tunarungu. Orang biasa sering sulit memahami bahasa isyarat yang digunakan oleh penyandang tunarungu. Masalah tersebut dapat mempengaruhi kesenjangan sosial antara penyandang disabilitas tunarungu dan masyarakat biasa.

Dari permasalahan tersebut penulis membuat penelitian tentang objek deteksi

penerjemah SIBI secara realtime dan non realtime. Tujuan dari penelitian ini untuk memahami proses objek deteksi dari hasil pelatihan model yang digunakan dan menganalisis optimasi parameter untuk mengetahui performa yang baik. Metode yang digunakan pada penelitian ini menggunakan metode literatur, studi lapangan dan perancangan simulasi. Berdasarkan metode tersebut dapat menjalankan penelitian ini dengan baik.

II. KAJIAN TEORI

A. SIBI (Sistem Isyarat Bahasa Indonesia)

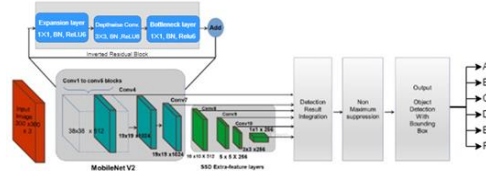
Sistem Isyarat Bahasa Indonesia (SIBI) merupakan salah satu media yang membantu komunikasi sesama kaum tunarungu di dalam masyarakat yang lebih luas. Wujudnya adalah tataan yang sistematis tentang seperangkat isyarat jari, tangan, dan berbagai gerak yang melambangkan kosa kata bahasa Indonesia. [1]

B. Deep Learning

Deep learning merupakan salah satu bidang machine learning yang memanfaatkan banyak layer pengolahan informasi nonlinier untuk melakukan ekstraksi fitur, pengenalan pola, dan klasifikasi. Secara teknis Deep Learning dapat diartikan teknik dalam Neural Network yang menggunakan teknik tertentu untuk mempercepat proses pembelajaran Neural Network yang menggunakan lapis yang banyak atau lebih dari 7 lapis. [2]

C. SSD Mobilnet V2

Dalam pembuatan sistem objek deteksi, diperlukan pelatihan model *deep learning* menggunakan algoritma *SSD MobileNet V2*. Adapun diagram dari algoritma tersebut dijabarkan pada Gambar 2.1 berikut.



GAMBAR 2.1
SSD MOBILENET V2

Gambar 2.1 menunjukkan representasi skematis algoritma deteksi objek SSD MobileNet V2. Ekstraktor fitur SSD mengekstrak fitur tingkat tinggi dari aliran gambar yang diambil dan menghasilkan peta fitur. Peta fitur diperlukan untuk tugas

membantu aliran pelatihan model. Model deteksi, SSD menggunakan peta fitur dan mendeteksi kelas suatu objek dan lokasinya menggunakan kotak pembatas. Kerangka pada gambar 3.4 dilatih untuk mengidentifikasi 6 kelas huruf SIBI (A, B, C, D, E, dan F)

Proses MobileNet v2 dimulai dengan ekstraksi fitur menggunakan Conv2D dengan kernel 3x3 untuk input gambar. Kemudian hasil dari Conv2D masuk ke 3 lapisan MobileNet V2. Lapisan ini terdiri dari tiga operasi konvolusi. Ada *Expansion layer* 1x1, *depthwise convolution* 3x3, dan *linear bottleneck convolution* 1x1. *Expansion layer* 1x1 meningkatkan jumlah saluran untuk memperkaya fitur. Lapisan *depthwise convolution* dan *linear bottleneck convolution* bertujuan untuk mengurangi biaya komputasi dan jumlah parameter. Ketiga lapisan ini ditambahkan diakhir dengan *inverted residual block* untuk mempermudah aliran gradien dalam pelatihan model jaringan. [3]

Proses SSD dengan pelokalan objek berjalan di SSD Feature Extractor. Jaringan menggunakan output dari beberapa lapisan terakhir (peta fitur) dari ekstraktor fitur untuk memprediksi lokasi objek dan kelas. Dalam arsitektur SSD, input dirutekan melalui lapisan konvolusi yang berbeda dengan ukuran yang berbeda. Lapisan ini menjadi semakin kecil di seluruh jaringan. Tujuan dari lapisan ini adalah untuk memungkinkan jaringan untuk mengenali objek dari berbagai bentuk dan ukuran. Satu set prediksi tetap diambil dari masing-masing lapisan konvolusi ini dan akhirnya digabungkan. Kedua, memilih bidang set-default mengurangi beban komputasi model. Model ini menciptakan lokasi dan klasifikasi dari berbagai objek dalam gambar. [4]

Semua output dari kotak prediksi objek digabungkan menjadi satu set. Kelompok disaring dengan penekanan non-maksimal dengan batas IoU >0,5. Deteksi objek kemudian memprediksi objek untuk kelas non-real-time dan real-time A, B, C, D, E, F. [3]

D. Parameter perofamansi

Parameter performansi adalah nilai parameter yang digunakan untuk mempengaruhi proses pembelajaran. Dalam menyetel parameter model dapat meningkatkan kinerja lebih jauh lagi. Penyetelan parameter biasanya disebut

sebagai transfer learning. Berikut contoh parameter performansi.

1. Model Arsitektur

Model arsitektur adalah model neural network yang digunakan untuk pelatihan deep learning. Model Arsitektur yang digunakan pada penelitian ini yaitu SSD MobileNet v2. SSD MobileNet terdiri dari SSD yang berperan sebagai base model, dan MobileNet v2 sebagai Network Model. Penggabungan SSD dan MobileNet akan membantu dalam proses pelatihan model deteksi objek.

2. Batch size

Batch size adalah hyperparameter yang menentukan jumlah sampel yang harus dikerjakan sebelum memperbarui parameter model internal. Sebuah dataset pelatihan dapat dibagi menjadi satu atau lebih batch size. Dalam penelitian ini menggunakan batch size 4 (mini-batch gradient descent). [5]

3. Epoch

Jumlah epoch adalah hyperparameter yang menentukan berapa kali algoritma pembelajaran akan bekerja melalui seluruh dataset pelatihan. Satu epoch berarti bahwa setiap sampel dalam set data pelatihan memiliki kesempatan untuk memperbarui parameter model internal.[5] Dalam penelitian ini menggunakan jumlah epoch 1.

4. Steps

Steps merupakan jumlah langkah dalam pelatihan. Langkah pelatihan adalah satu pembaruan gradien. Istilahnya satu langkah batch size diproses. Semakin besar jumlah steps maka semakin banyak waktu yang dibutuhkan pada proses pelatihan.

5. Learning rate

Learning rate adalah kecepatan dalam mempeleajari jaringan. Nilai yang terdapat *learning rate* bernilai antara 0 sampai 0,9. Jika nilai *learning rate* nya terlalu kecil maka proses *training* membutuhkan waktu yang lama, hal ini dikarenakan *epoch* yang dibutuhkan semakin banyak untuk mencapai nilai target yang diinginkan. Semakin besar nilai *learning rate* yang digunakan maka proses pelatihan jaringan akan semakin cepat, namun jika terlalu besar akan mengakibatkan jaringan tersebut tidak stabil dan menyebabkan nilai *error* berulang pada nilai tertentu. [6]

6. Loss

Fungsi *loss* yang digunakan dalam SSD adalah kombinasi dari *localization loss* dan *classification loss*. *localization loss* adalah *smooth loss* L1 yang didefinisikan dengan cara berikut: [7]

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 0.5 \\ |x| - 0.5 & \text{sebaliknya} \end{cases} \quad (2.1)$$

$$L_{loc}(x, l, g) = \sum_N \sum_{m \in \{cx, cy, w, h\}} x_{ij} smooth_{L1}(l_i^m - g_j^m) \quad (2.2)$$

Smooth loss dihitung untuk semua kotak pembatas default N antara parameter kotak pembatas yang diprediksi (l) dan kotak pembatas kebenaran dasar (g). *classification loss* adalah kehilangan lintas entropi sederhana atas skor kelas yang dihitung softmax untuk kelas. *classification loss* diberikan oleh persamaan berikut: [7]

$$L_{conf}(x, c) = - \sum_N x_{ij} \log(\hat{c}_i^p) - \sum_N \log(\hat{c}_i^0) \quad (2.4)$$

Istilah \hat{c}_i^p adalah jika kotak pembatas berisi objek dan istilah \hat{c}_i^0 adalah ketika kotak pembatas tidak ada objek. \hat{c} mewakili skor softmax kelas.

Fungsi total kerugian dihitung sebagai berikut: [7]

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \quad (3.4)$$

N adalah jumlah kecocokan positif dan α adalah bobot untuk kehilangan lokalisasi yang diset 1 dengan cross validation.

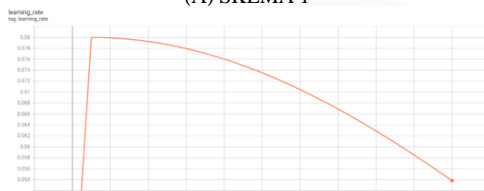
7. Average Precision & Average Recall

AP (Average Presisi) adalah metrik populer dalam mengukur tingkat klasifikasi detektor objek. AP menghitung nilai presisi rata-rata untuk nilai recall lebih dari 0 hingga 1. Berikut beberapa parameter yang memengaruhi AP yaitu Precision dan Recall. Presisi adalah mengukur seberapa akurat prediksi dengan persentase prediksi benar. Recall adalah mengukur seberapa baik untuk menemukan semua positif. Presisi dan recall mengukur berapa banyak prediksi yang dibuat model sebenarnya. [8] AP dirata-ratakan untuk semua kategori. Secara tradisional, ini disebut "rata-rata presisi rata-rata" (mAP). AR (Average Recall) adalah penarikan maksimum yang diberikan

Pada pelatihan akan membentuk suatu kecepatan dalam mempelajari dan melatih jaringan yang disebut *learning rate*. Satuan dari *Learning rate* adalah *iteration/steps*. *Learning rate* pada penelitian ini di bagi menjadi 3 skema yaitu skema 1 pelatihan jaringan dengan jumlah 10.000 steps dengan rata rata *learning rate* sebesar 0,075 *iteration/steps*. Skema 2 pelatihan jaringan dengan jumlah 20.000 steps dengan rata-rata *learning rate* 0,070 *iteration/steps* dan skema 3 pelatihan jaringan dengan jumlah 30.000 steps dengan rata-rata *learning rate* 0,060 *iteration/steps*. Ketiga skema ini ditinjau pada *learning rate* yang didapatkan pada tensorboard sebagai visualisasi data. Berikut **Gambar 3.1(A)**, **Gambar 3.1(B)**, dan **Gambar 3.1(C)** grafik yang menunjukan *learning rate* ketiga skema



GAMBAR 3.1
(A) SKEMA 1

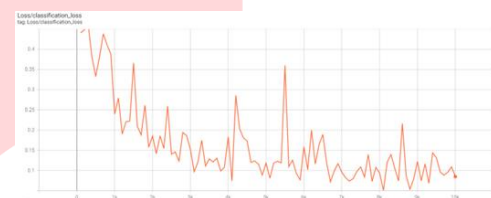


GAMBAR 3.1
(B) SKEMA 2



GAMBAR 3.1
(C) SKEMA 3

gradient descent berusaha untuk menemukan keseimbangan antara ketahanan *stochastic gradient descent* dan efisiensi *batch gradient descent* sehingga menghasilkan anomali pada saat pelatihan jaringan. Pada penelitian *classification loss* dinilai dari 3 skema. Skema 1 yaitu pelatihan jaringan dengan jumlah steps 10.000 dengan rata-rata *classification loss* sebesar 0,1627, skema 2 yaitu pelatihan jaringan dengan jumlah steps 20.000 dengan rata-rata *classification loss* sebesar 0,1294 dan skema yaitu pelatihan jaringan dengan jumlah steps 30.000 dengan rata-rata *classification loss* sebesar 0,1111. Berikut **Gambar 3.2(A)**, **Gambar 3.2(B)**, dan **Gambar 3.2(C)** grafik yang menunjukan *classification loss* ketiga skema.



GAMBAR 3.2
(A) SKEMA 1



GAMBAR 3.2
(B) SKEMA 2



GAMBAR 3.2
(C) SKEMA 3

B. Classification loss

Classification loss merupakan kerugian klasifikasi objek dalam tahap pelatihan jaringan. Secara teknis *classification loss* merupakan kehilangan lintas entropi sederhana atas skor kelas yang dihitung softmax untuk kelas. *Classification loss* dipengaruhi mini batch gradient descent yang membuat algoritma penurunan gradient membagi dataset menjadi kumpulan kecil untuk menghitung kesalahan model dan memperbarui koefisien model. Mini batch

C. Localization loss

localization loss merupakan kerugian dalam memprediksi offset kotak pembatas untuk setiap objek. Secara teknis *localization loss* merupakan kerugian yang dihitung untuk semua kotak pembatas default, antara parameter kotak pembatas yang diprediksi dan kotak pembatas kebenaran dasar. Selain itu *localization loss* dipengaruhi mini batch gradient descent yang membuat algoritma penurunan gradient membagi dataset menjadi

kumpulan kecil untuk menghitung kesalahan model dan memperbarui koefisien model. Mini batch gradient descent berusaha untuk menemukan keseimbangan antara ketahanan *stochastic gradient descent* dan efisiensi *batch gradient descent* sehingga menghasilkan anomali pada saat pelatihan jaringan. Pada penelitian ini localization loss dilakukan pada 3 skema. Skema 1 yaitu pelatihan jaringan dengan jumlah steps 10.000 dengan rata-rata localization loss sebesar 0,0857, skema 2 yaitu pelatihan jaringan dengan jumlah steps 20.000 dengan rata-rata localization loss sebesar 0,0639 dan skema yaitu pelatihan jaringan dengan jumlah steps 30.000 dengan rata-rata localization loss sebesar 0,0519. Berikut **Gambar 3.3(A)**, **Gambar 3.3(B)**, dan **Gambar 3.3(C)** grafik yang menunjukkan localization loss ketiga skema.



GAMBAR 3.3 (A) SKEMA 1



GAMBAR 3.3 (B) SKEMA 2



GAMBAR 3.2 (C) SKEMA 3

D. Total loss

Total loss merupakan total kerugian dari hasil pelatihan jaringan yang terdiri 3 skema. Skema 1 yaitu pelatihan jaringan dengan jumlah steps 10.000 dengan rata-rata total loss sebesar 0,3926, skema 2 yaitu pelatihan jaringan dengan jumlah steps 20.000 0,3209 dan skema yaitu pelatihan jaringan dengan jumlah steps 30.000 dengan rata-rata total

loss sebesar 0,2750. Berikut **Gambar 3.4(A)**, **Gambar 3.4(B)**, dan **Gambar 3.4(C)** grafik yang menunjukan total loss ketiga skema



GAMBAR 3.4 (A) SKEMA 1



GAMBAR 3.4 (B) SKEMA 2



GAMBAR 3.4 (C) SKEMA 3

E. mAP & mAR

Parameter pada evaluasi yang merupakan *coco evaluation metric* terdiri mAP (mean Average Precision) dan mAR (mean Average Recall) dalam mendeteksi objek. Nilai rata-rata mAP pada seluruh skema sebesar 0,75 dan nilai rata-rata mAR pada seluruh skema sebesar 0,78. mAP dan mAR didapatkan menggunakan tensorboard sebagai data visualisasi. Dari data tersebut mendapatkan jenis-jenis mAP dan mAR pada keseluruhan gambar dan kelas. Berikut **Tabel 3.1** yang menunjukkan mAP dan **Tabel 3.2** mAR pada seluruh skema.

TABEL 3.1 MAP

AP (Average Precision)	Ske ma 1	Ske ma 2	Ske ma 3	Rata-rata
mAP	0,7376	0,753	0,7594	0,75
mAP large	0,7477	0,76	0,76	0,7559
mAP medium	0,7257	0,7682	0,7893	0,76106667
mAP small	-1	-1	-1	-1

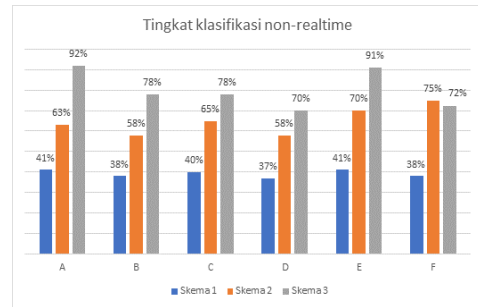
mAP@.5 0 IoU	1	0,98 73	0,99 01	0,99246 6667
mAP@.7 5IoU	0,93 42	0,96 78	0,96 06	0,9542

TABEL 3.2
MAR

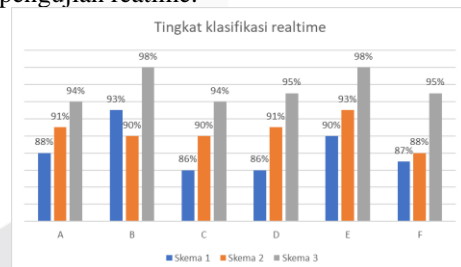
AR (Average Recall)	Ske ma 1	Ske ma 2	Ske ma 3	Rata- rata
mAR @1	0,77 33	0,79 08	0,79 17	0,78526 6667
mAR @10	0,77 5	0,79 25	0,79 17	0,7864
mAR @100	0,77 5	0,79 25	0,79 25	0,78666 6667
mAR @100 large	0,78 2	0,79 95	0,79 26	0,79136 6667
mAR @100 mediu m	0,74 14	0,77 5	0,80 08	0,7724
mAR @100 small	-1	-1	-1	-1

F. Pengujian non-reatime dan realtime

Pengujian non realtime merupakan pengujian dengan mengunggah gambar kedalam pemrograman deteksi objek. Pengujian dilakukan dengan 20 gambar setiap kelasnya. Setiap gambar akan menghasilkan tingkat klasifikasi. Dengan adanya tingkat klasifikasi kita dapat mengetahui, seberapa besar prediksi deteksi dalam mendeteksi suatu objek. Rata-rata tingkat klasifikasi pada pengujian non realtime sebesar 60%. Nilai klasifikasi tertinggi diraih kelas E pada skema 3 sebesar 91 %. Nilai tingkat klasifikasi terendah didapat pada kelas D dengan skema 1 sebesar 37 %. Berikut **Gambar 3.5** yang menunjukkan grafik tingkat klasifikasi pada pengujian non-realtime.

GAMBAR 3.1
GRAFIK TINGKAT KLASIFIKASI NON-
REALTIME

Pengujian realtime merupakan pengujian deteksi objek dengan waktu yang nyata (realtime) menggunakan OpenCV untuk menangkap objek dengan kamera laptop. Proses deteksi objek realtime dilakukan dalam pemrograman yang diproses dari frame kamera yang menangkap gambar secara langsung. Kemudian diolah oleh model prediksi deteksi akan membuat deteksi pada gambar tangkapan tersebut kemudian ditampilkan kembali pada frame kamera tersebut beserta kotak, kelas tingkat klasifikasi deteksinya. Pengujian ini diambil 10 sampel gambar deteksi objek disetiap kelas dan skema. Berikut **Gambar 3.6** yang menunjukkan grafik tingkat klasifikasi pada pengujian realtime.

GAMBAR 3.2
GRAFIK TINGKAT KLASIFIKASI REALTIME

V. KESIMPULAN

Berdasarkan hasil analisis yang telah dilakukan, maka dapat diperoleh beberapa kesimpulan sebagai berikut:

- Penelitian ini merupakan simulasi pendeteksi huruf SIBI (A, B, C, D, E, F) yang dilatih dengan Model SSD MobileNet V2 jumlah steps 10.000 (skema 1), 20.000 (skema 2), 30.000 (skema 3) dan total dataset 1200 gambar.
- Hasil analisis pengujian parameter output pada hasil pelatihan seperti learning rate maksimal dari seluruh skema sebesar 0,079, nilai *classification loss*, *localization loss*, *total loss* paling rendah terletak pada

- skema 3. Sehingga skema 3 merupakan skema terbaik dari segi parameter output pada pelatihan.
- C. Hasil analisis pengujian output pada hasil evaluasi terdapat nilai mAP dan mAR tertinggi diraih oleh skema 3 dengan masing masing nilai 0,7594 dan 0,7917. Berdasarkan hal tersebut skema 3 menjadi terbaik dalam tahap evaluasi.
- D. Hasil analisis pengujian deteksi non-realtime pada seluruh skema terdeteksi dengan nilai tingkat klasifikasi berkisaran 0 % - 99 %. Nilai rata rata tingkat klasifikasi pada deteksi non-realtime sebesar 41 %. Tingkat keberhasilan deteksi
- pada deteksi non-realtime sebesar 79,8 %. Berdasarkan hal tersebut deteksi non-realtime cukup baik dalam mendeteksi objek.
- E. Hasil analisis pengujian deteksi *real time* pada seluruh skema terdeteksi dengan nilai tingkat klasifikasi berkisaran 81 % - 90 %. Nilai rata rata tingkat klasifikasi pada deteksi realtime sebesar 91 %. Tingkat keberhasilan deteksi pada deteksi *real time* sebesar 100%. Berdasarkan hal tersebut deteksi *real time* baik dalam mendeteksi objek.

REFERENSI

- [1] Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia, "Sistem Isyarat Bahasa Indonesia (Profil)," Kementerian Pendidikan dan Kebudayaan Indonesia, 30 September 2020. [Online]. Available: <https://pmpk.kemdikbud.go.id/sibi/profil>. [Accessed 12 September 2022].
- [2] A. Ahmad, "Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning," 2017.
- [3] M. Sandler, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2019.
- [4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," 2016.
- [5] J. Brownlee, "What is the Difference Between a Batch and an Epoch in a Neural Network?," Machine Learning Mastery, 5 August 2022. [Online]. Available: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>. [Accessed 10 August 2022].
- [6] S. R. DEWI, "DEEP LEARNING OBJECT DETECTION PADA VIDEO MENGGUNAKAN TENSORFLOW DAN CONVOLUTIONAL NEURAL NETWORK," 2018.
- [7] Y. (. Liu and S. Mehta, Hands-On Deep Learning Architectures with Python, Birmingham: Packt Publishing Ltd., 2019.
- [8] J. Hui, "mAP (mean Average Precision) for Object Detection," medium, 7 May 2018. [Online]. Available: <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>. [Accessed 2 December 2021].
- [9] COCO Consortium, "Detection Evaluation," COCO Common Object in Context, [Online]. Available: <https://cocodataset.org/#detection-eval>.
- [10] S. Yohanandan, "mAP (mean Average Precision) might confuse you!," Towards Data Science, 9 June 2020. [Online]. Available: <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>. [Accessed 30 November 2021].
- [11] A. Wibowo, C. A. Hartanto and P. W. Wirawan, "Android skin cancer detection and classification based on MobileNet v2 model," International Journal of Advances in Intelligent Informatics, vol. 6, no. 2, pp. 135-148, July 2020.