Deteksi Rompi dan Helm Keselamatan Menggunakan Metode YOLO dan CNN

Rescky Marthen Mailoa, Leo Willyanto Santoso
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) - 8417658
E-Mail: risckymarthen@gmail.com, leow@petra.ac.id

ABSTRAK

Pekerja konstruksi dalam menjalankan tugasnya memiliki resiko yang tinggi dan rawan dengan kecelakaan. Beberapa faktor yang membuat resiko kecelakaan menjadi semakin besar yaitu kondisi lingkungan yang terbuka dan panas, ketinggian bangunan, berhadapan dengan benda-benda tajam, dan lainya. Penggunaan alat pelindung diri (APD) menjadi penting untuk mengantisipasi atau mengurangi resiko kecelakaan yang mungkin terjadi. Namun, tidak jarang para pekerja konstruksi lupa atau mungkin sengaja tidak menggunakan alat pelindung diri. Untuk mengatasi permasalahan tersebut dibutuhkan sebuah sistem yang dapat mendeteksi alat pelindung diri pada pekerja konstruksi.

Penelitian ini menggunakan metode *You Only Look Once* (YOLO) untuk mendeteksi bagian tubuh kepala dan badan dari gambar yang di-inputkan. Bagian tubuh yang terdeteksi kemudian dipotong dan diproses dengan metode *Convolutional Neural Network* (CNN) dengan *model ResNet50* untuk diklasifikasikan. Proses pelatihan dengan *model ResNet50* akan dilakukan modifikasi pada *hyperparameter* seperti *learning rate, epoch, dense layer, dropout layer,* augmentasi data dan *freeze layer* untuk membandingkan performanya dengan *model* sebelum dilakukan modifikasi.

Hasil penelitian menunjukan model YOLO memiliki tingkat kecepatan mendeteksi yang sangat tinggi dengan akurasi yang baik. Sementara itu *model* CNN yang dimodifikasi menunjukan performa yang baik dengan nilai akurasi rata-rata 96%.

Kata Kunci: *You Only Look Once, Convolutional Neural Network, Residual Network, Hyperparameter*, Klasifikasi Gambar, Alat Pelindung Diri

ABSTRACT

Construction workers face a high risk of injury and are prone to accidents while performing their duties. Several factors increase the chance of accidents, including open and heated environmental conditions, building heights, sharp objects, and others. The use of personal protective equipment (PPE) is essential to anticipate or reduce the risk of accidents that may occur. However, it is not uncommon for construction workers to forget or purposefully disregard personal protective equipment. To address these problems, a system capable of detecting personal protective equipment for construction workers is required.

This study used the You Only Look Once (YOLO) method to detect the head and body parts of the inputted image. The detected body parts were then cut and processed using the Convolutional Neural Network (CNN) method with the ResNet50 model for classification. The training process with the ResNet50 model was modified on hyperparameters including learning rate, epoch, dense layer, dropout layer, data augmentation, and freeze layer to compare its performance with the model before modification.

The results showed that the YOLO model has a very high level of detection speed with good accuracy. Meanwhile, the modified CNN model performed well with an average accuracy value of 96%.

Keywords: *You Only Look Once, Convolutional Neural Network, Residual Network, Hyperparameter, Image Classification, Personal Protective Equipment*.

1. PENDAHULUAN

Pekerja konstruksi merupakan salah satu unsur utama yang sangat berperan penting dalam menjalankan sebuah kegiatan konstruksi. Dalam menjalankan tugasnya, para pekerja konstruksi memiliki resiko yang tinggi dan rawan dengan kecelakaan. Beberapa faktor yang membuat resiko kecelakaan menjadi semakin besar seperti kondisi lingkungan yang terbuka dan panas, ketinggian bangunan, berhadapan dengan benda-benda tajam, dan lain-lain. Oleh karena itu, penggunaan alat pelindung diri (APD) sangatlah penting untuk mengantisipasi atau mengurangi resiko kecelakaan yang mungkin terjadi ketika para pekerja konstruksi menjalankan tugasnya.

Alat pelindung diri (APD) adalah seperangkat alat yang sering digunakan oleh para tenaga kerja untuk melindungi tubuhnya dari potensi bahaya dan kecelakaan kerja. Namun dalam menjalankan tugasnya, tidak jarang para pekerja konstruksi lupa atau mungkin sengaja tidak menggunakan alat pelindung diri karena ketidaknyamanan yang disebabkan oleh berat dan membuat suhu tubuh mereka menjadi semakin panas [10]. Hal inilah yang dianggap sangat berbahaya dan bisa memperbesar resiko kecelakaan kerja yang mungkin akan terjadi. Untuk mengatasi masalah ini, dibutuhkan sebuah sistem yang dapat mendeteksi apakah seorang pekerja konstruksi menggunakan alat pelindung diri atau tidak.

Deteksi alat pelindung diri dalam beberapa tahun terakhir telah mendapatkan perhatian yang cukup signifikan. Hal ini dianggap penting karena sangat diperlukan dalam manajemen keselamatan dan produktivitas para pekerja konstruksi [8]. Selain itu, hal ini juga bisa dimanfaatkan untuk memonitor pekerja dan menyoroti operasi atau kegiatan yang tidak aman untuk mengantisipasi kecelakaan yang mungkin bisa terjadi [4]. Dengan demikian, deteksi alat pelindung diri yang terus dikembangkan ini bisa menjadi salah satu solusi yang cukup menjawab permasalahan ini.

Pada penelitian ini, diharapkan penggunaan metode YOLO dan CNN dapat membangun sistem yang bisa membantu dalam mendeteksi alat pelindung diri khususnya rompi dan helm keselamatan pada para pekerja konstruksi sehingga resiko kecelakaan kerja bisa semakin berkurang.

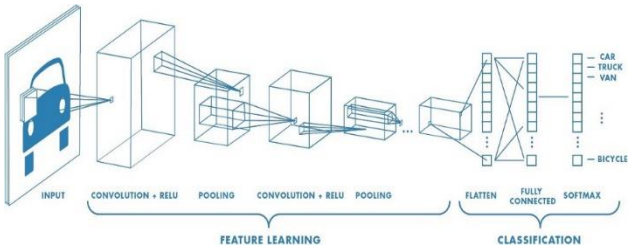
2. LANDASAN TEORI

2.1 Alat Pelindung Diri

Alat pelindung diri (APD) merupakan salah satu komponen penting yang dibutuhkan oleh para pekerja untuk menjaga keamanan dan keselamatan mereka di lingkungan kerja yang penuh dengan berbagai ancaman dan risiko. APD inilah yang akan digunakan untuk meminimalkan risiko dan potensi bahaya di lingkungan kerja [2].

2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu komponen atau bagian dari *neural network* yang dirancang untuk memproses data berupa gambar. Gambar yang diinputkan akan diproses melalui *layer-layer* dari CNN untuk diekstrak informasi dari gambar yang diterima sehingga nantinya gambar tersebut dapat dikenali dan diklasifikasikan. CNN tersusun dari tiga *layer* utama, yaitu *convolution layer*, *pooling layer* dan *fully-connected layer*. Arsitektur dari CNN dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur *Convolutional Neural Network*

Convolutional layer adalah *layer* pertama dan merupakan inti dari arsitektur CNN. Pada *layer* ini akan dilakukan proses konvolusi *filter* berupa matriks yang umumnya berukuran 3x3 untuk mengekstrak fitur yang menonjol dari gambar yang diinputkan. Selain itu, terdapat juga *stride* dan *padding* yang juga berperan penting pada *layer* ini. *Stride* merupakan parameter yang akan menentukan jumlah pergeseran *filter* pada matriks. *Padding* atau *zero padding* merupakan teknik yang digunakan untuk mempertahankan ukuran asli dari gambar yang diinputkan.

Pada *pooling layer* akan dilakukan pengecilan matriks dengan cara mengambil bagian atau kelompok dari fitur yang dihasilkan pada *convolutional layer* [7]. Fitur yang diambil kemudian akan diproses hingga menjadi satu nilai yang akan merepresentasikan nilai-nilai dari kelompok atau bagian yang dipilih.

Setelah melewati *pooling layer*, terdapat *fully connected layer* yang merupakan *layer* terakhir dari arsitektur CNN. Pada *layer* ini akan dilakukan klasifikasi dari *input* yang diterima. Sebelum melakukan klasifikasi, *feature* yang dihasilkan pada *layer-layer* sebelumnya harus di *reshape* atau *flatten* menjadi sebuah vektor agar bisa digunakan sebagai *input* dari *fully connected layer*. Pada bagian akhir akan ada fungsi aktivasi seperti *softmax* atau *sigmoid* untuk mengklasifikasikan berdasarkan nilai kategori tertinggi.

2.3 You Only Look Once

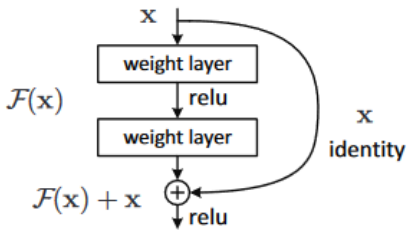
You Only Look Once (YOLO) merupakan salah satu metode yang sering digunakan untuk melakukan deteksi objek pada sebuah gambar. Proses deteksi objek akan dilakukan dengan menentukan terlebih dahulu lokasi dari setiap objek yang ingin dideteksi pada gambar. Dimana YOLO merupakan sebuah metode yang menggunakan penerapan jaringan saraf tunggal *(single neural network)* pada keseluruhan gambar. Jaringan inilah yang nantinya akan digunakan untuk membagi gambar menjadi wilayah-wilayah *(grid)* dan kemudian akan dilakukan proses prediksi dari setiap wilayah *(bounding box)* untuk selanjutnya akan diklasifikasikan sebagai objek atau bukan objek [6]. Alur proses deteksi YOLO dapat dilihat pada Gambar 2.

Gambar 2. Alur proses deteksi pada YOLO

YOLO akan membagi gambar inputan menjadi *grid* berukuran S x S. Setiap *grid sel* akan memprediksi *bounding box* beserta nilai dari *bounding box* tersebut. Atribut yang akan dihasilkan antara lain, 4 koordinat (x, y, w dan h) dari *bounding box* beserta *confidence score* yang menunjukan nilai probabilitas adanya objek pada posisi tersebut. Atribut x dan y menunjukan posisi pusat dari *bounding box*. Sedangkan w dan h menggambarkan lebar dan tinggi perkiraan objek pada gambar. *Confidence score* akan dihitung dengan mengalikan probabilitas keberadaan objek pada *grid* dengan IOU *(Intersection Over Union) bounding box* prediksi dengan IOU *ground truth* nya.

2.4 Deep Residual Learning

Deep Residual Network atau ResNet adalah salah satu arsitektur dari *neural network*. Arsitektur ini diciptakan untuk menyelesaikan masalah dimana saat dilakukan perbandingan antara *network* dengan *layer* 20 dan *layer* 56 menunjukan hasil dari *network* dengan *layer* 56 memiliki *training error* dan *testing error* yang lebih tinggi dibandingkan dengan *layer* 20. Berangkat dari permasalahan diatas *Kaimin He, Xiangyu Zhang, Shaoqin Ren* dan *Jian Sun* pada tahun 2015 mengusulkan sebuah solusi dimana akan ditambahkan *skip connection* atau *shortcut connection* yang bekerja dengan melewati beberapa lapisan pada *neural network* dan mengumpankan *output* dari satu lapisan sebagai *input* pada lapisan berikutnya. *Skip connection* bekerja dengan membiarkan lapisan jaringan memperoleh transformasi fitur untuk ditambahkan ke fitur itu sendiri [3], seperti ditunjukan pada Gambar 3.



Gambar 3. Proses b*uilding block* pada *residual learning*

Pada penelitian ini akan digunakan *ResNet50* yang memiliki 50 *layer*. Selain *ResNet50*, terdapat beberapa tipe atau varian arsitektur ResNet lainnya seperti *ResNet18*, *ResNet34*, *ResNet101* dan *ResNet152*. ResNet juga tersusun dari *convolutional layer*, *pooling layer*, dan *fully connected layer*. Pada bagian akhir, ResNet menggunakan *softmax* sebagai fungsi aktivasi.

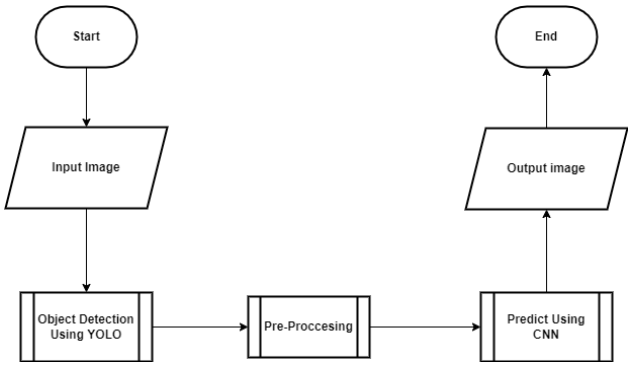
3. DESAIN SISTEM

3.1 Analisis Data

Data yang akan digunakan pada penelitian ini dikumpulkan dari dua sumber *dataset* yang berbeda. Sumber yang pertama diambil dari *Hardhat and Safety Vest Image for Object Detection* oleh John Syin, data yang digunakan sebanyak 2186 gambar [9]. Untuk sumber yang kedua menggunakan data dari *Helmet_Dataset* oleh Abhishek Annamraju, data yang digunakan sebanyak 552 gambar [1].

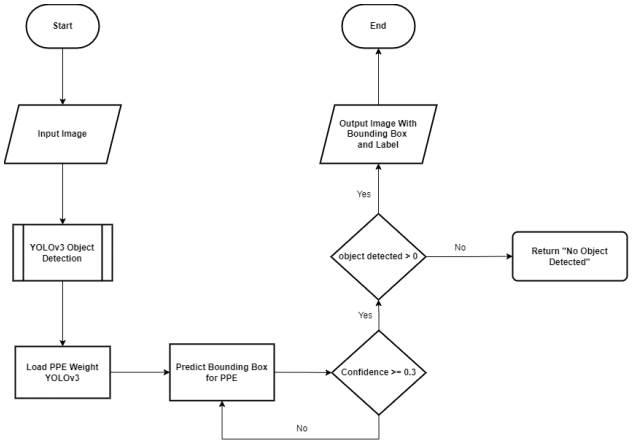
3.2 Analisis Sistem

Sistem pertama-tama akan menerima *input* data berupa gambar dengan format *.jpg*. Gambar yang diterima kemudian akan diolah lebih lanjut untuk dideteksi apakah ada badan ataupun kepala didalamnya atau tidak. Proses deteksi objek ini akan dilakukan dengan menggunakan metode YOLO untuk menentukan posisi atau lokasi dari badan dan kepala yang ada pada gambar yang diinputkan. Setelah berhasil dideteksi menggunakan YOLO selanjutnya akan dilakukan proses klasifikasi menggunakan metode CNN. Arsitektur dapat dilihat pada Gambar 4.



Gambar 4. Arsitektur sistem

3.2.1 You Only Look Once

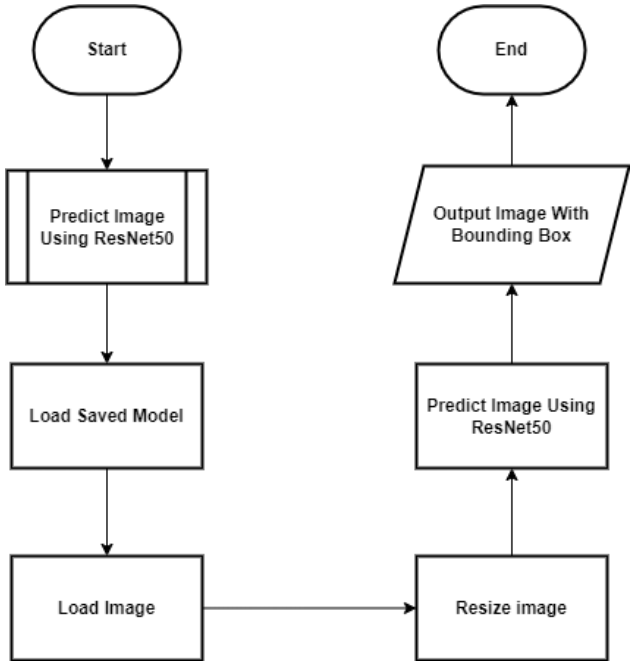


Gambar 5. *Flowchart* Proses *You Only Look Once*

You Only Look Once (YOLO) akan dimanfaatkan untuk mendeteksi objek pada gambar yang diinputkan. Objek yang akan dideteksi merupakan 2 bagian tubuh yaitu kepala dan badan dari gambar yang diinputkan. Proses deteksi ini akan dijalankan dengan memanfaatkan arsitektur dari YOLOv3. Karena memiliki 53 konvolusional *layer* maka arsitektur ini dapat disebut juga sebagai *Darknet-53* [5]. Setelah berhasil mendapatkan objek dengan nilai *confidence* tertinggi, akan dilakukan pemotongan. Hasil potongan inilah yang nantinya akan digunakan pada bagian *convolutional neural network* untuk diolah lebih lanjut. Gambar 5 merupakan *flowchart* proses pada YOLO.

3.2.2 Convolutional Neural Network

Pada penelitian ini akan dimanfaatkan salah satu arsitektur dari *Convolutional Neural Network* yaitu *Deep Residual Network* (ResNet50). Sebelum diproses pada ResNet50 gambar yang akan diolah harus di-*resize* terlebih dahulu. Hal ini dikarenakan inputan yang diterima pada ResNet50 adalah 224x224 *pixel*. Setelah gambar telah berhasil di-*resize* maka gambar telah siap untuk diproses menggunakan *ResNet50*. Hasil dari proses ini akan mendapatkan *output* berupa klasifikasi dari objek yang telah dideteksi pada gambar yang diinputkan. *Flowchart* dari ResNet dapat dilihat pada Gambar 6.



Gambar 6. *Flowchart* Proses *ResNet50*

3.3 Desain Sistem

Desain sistem yang ditampilkan adalah *user interface* dari program yang dikerjakan. Pada bagian awal, akan ada tempat yang dapat digunakan oleh *user* untuk memilih dan meng-*upload* gambar yang nantinya akan diproses untuk dideteksi rompi dan helm keselamatan di dalamnya. Pada bagian ini juga, akan ditampilkan hasil deteksi rompi dan helm keselamatan dari gambar yang telah diinputkan tersebut. *Input* yang diterima pada bagian ini hanya *file* dengan ekstensi *.jpg*.

4. IMPLEMENTASI SISTEM

Sistem yang dibuat akan diimplementasikan pada komputer dengan *operating system* Windows 10 dengan spesifikasi GPU *NVIDIA GeForce GTX 950M* dan RAM 12gb. Bahasa pemrograman yang digunakan adalah *python* 3.8. Penelitian ini dibantu dengan *framework tensorflow* 2.2. Selain itu, terdapat beberapa *library* yang digunakan untuk mendukung sistem ini antara lain *streamlit, CV2, numpy, scikit-learn* dan *matplotlib*.

5. ANALISA DAN PENGUJIAN

5.1 Pengujian Sistem

Pengujian sistem dilakukan dengan mengevaluasi performa dari sistem *training* dan *testing* yang telah dibuat. Pengujian ini akan dilakukan pada masing-masing metode yang digunakan. Pencatatan akan dibedakan berdasarkan jenis alat pelindung diri yang diprediksi dan diklasifikasikan. Pada metode YOLO kelas yang akan dideteksi adalah kepala dan badan. Sementara pada metode CNN kelas yang akan diklasifikasikan adalah *helmet*, *no_helmet*, *no_vest* dan *vest*.

5.1.1 Pengujian Metode YOLOv3

Pengujian dengan metode YOLO dilakukan untuk mendeteksi kepala dan badan. Jumlah iterasi atau *epoch* yang digunakan adalah 4000 dengan nilai *learning rate* 0.001. Pengujian ini dilakukan untuk memperoleh nilai akurasi dan *recall* dari proses deteksi. Proses perhitungan akurasi dan *recall* akan menggunakan persamaan (1) dan (2). Hasil perhitungan dapat dilihat pada Tabel 1, Tabel 2 dan Tabel 3.

$$Akurasi = \frac{Jumlah\ objek\ terdeteksi}{Jumlah\ objek\ keseluruhan} \times 100 \qquad (1)$$

Tabel 1. Hasil pengujian untuk kelas kepala

Confidence	Jumlah Kepala Terdeteksi	Jumlah Kepala Keseluruhan	Akurasi
0.3	**1103**	**1721**	**64.09%**
0.4	598	1721	34.75%
0.5	192	1721	11.16%

Tabel 2. Hasil pengujian untuk kelas badan

Confidence	Jumlah badan Terdeteksi	Jumlah badan Keseluruhan	Akurasi
0.3	**774**	**1228**	**63.03%**
0.4	404	1228	32.90%
0.5	120	1228	9.77%

Hasil pengujian akurasi pada kelas kepala dan badan menunjukan performa yang lebih baik pada pengujian dengan nilai *confidence* 0.3. Dimana hasil akurasi yang diperoleh untuk kelas kepala sebesar 64.09% dan untuk kelas badan akurasi yang diperoleh sebesar 63.03%.

$$Recall = \frac{Jumlah\ gambar\ terdeteksi}{Jumlah\ gambar\ keseluruhan} \times 100 \qquad (2)$$

Tabel 3. Hasil pengujian recall dengan nilai confidence 0.3

Class	Jumlah Gambar	Gambar Terdeteksi	*Recall*
Kepala	552	487	88.22%
Badan	552	486	88.04%

Hasil pengujian recall menunjukan performa yang cukup baik dengan berhasil mendeteksi 487 gambar dari total 552 pada kelas kepala dan berhasil mendeteksi 486 gambar dari total 552 pada kelas badan.

5.1.2 Pengujian Model ResNet50

Pengujian pada *model ResNet50* dilakukan dengan memanfaatkan proses *transfer learning* pada *model ResNet50* yang sudah dilatih terlebih dahulu. Kemudian *model* akan dilakukan *fine-tuning* pada beberapa *parameter* utamanya agar *model* mampu memberikan hasil yang lebih baik pada *dataset* yang akan di-*training*.

5.1.2.1 Pengujian Learning Rate

Pengujian akan dilakukan dengan jumlah *epoch* 100 dan *batch size* 32. Nilai *learning rate* yang akan diujikan adalah 1e-3, 1e-4 dan 1e-5. Hasilnya ditunjukan pada Tabel 4.

Tabel 4. Hasil pengujian *learning rate*

Learning rate	*Avg Train Accuracy*	*Avg Train Loss*	*Avg Val Accuracy*	*Avg Val Loss*
1e-3	99.17%	0.02	94.64%	0.47
1e-4	**99.17%**	**0.01**	**95.54%**	**0.42**
1e-5	99.03%	0.02	93.75%	0.49

Tabel 4 menunjukan hasil *learning rate* dengan nilai 1e-4 memiliki performa lebih baik dibandingkan 2 nilai lainnya yang diujikan.

5.1.2.2 Pengujian Jumlah Epoch

Pengujian akan dilakukan dengan nilai *learning rate* 1e-4 dan *batch size* 32. Jumlah *epoch* yang digunakan pada pengujian ini adalah 150. Hasilnya ditunjukan pada Tabel 5.

Tabel 5. Hasil pengujian jumlah *epoch*

Epoch	*Avg Train Accuracy*	*Avg Train Loss*	*Avg Val Accuracy*	*Avg Val Loss*
10	97.04%	0.08	93.30%	0.44
20	98.98%	0.03	95.54%	0.38
30	98.80%	0.03	94.64%	0.37
40	98.47%	0.04	92.41%	0.52
50	98.66%	0.06	94.64%	0.69
60	99.17%	0.02	94.20%	0.67
70	99.12%	0.02	94.20%	0.62
80	99.17%	0.02	95.09%	0.46
90	98.94%	0.02	94.64%	0.56
100	**99.17%**	**0.01**	**95.54%**	**0.42**
110	98.52%	0.02	90.62%	0.90
120	99.03%	0.01	95.98%	0.45
130	99.03%	0.01	95.98%	0.49
140	99.03%	0.01	95.98%	0.49
150	98.84%	0.02	92.86%	0.59

Hasil pengujian jumlah *epoch* dengan perubahan tiap 10 *epoch* menunjukan nilai *training loss* dan *validation loss* paling minimal diperoleh pada saat *epoch* 100.

5.1.2.3 Pengujian Dropout Layer

Pengujian akan dilakukan dengan nilai *learning rate* 1e-4, *batch size* 32 dan *epoch* 100. Nilai *rasio dropout* yang akan diujikan adalah 0.3, 0.4 dan 0.5. Hasilnya ditunjukan pada Tabel 6.

Tabel 6. Hasil pengujian *dropout layer*

Dropout	*Avg Train Accuracy*	*Avg Train Loss*	*Avg Val Accuracy*	*Avg Val Loss*
0.3	**99.26%**	**0.01**	**95.09%**	**0.37**
0.4	99.17%	0.01	95.09%	0.84
0.5	99.12%	0.02	94.20%	0.39

Tabel 6 menunjukan hasil *dropout* dengan nilai rasio 0.3 memperoleh nilai *training loss* dan *validation loss* paling minimal.

5.1.2.4 Pengujian Dense Layer

Pengujian akan dilakukan dengan nilai *learning rate* 1e-4, *dropout* 0.5 dan *epoch* 100. Nilai *unit* dari *dense layer* yang akan diujikan adalah 256, 512 dan 1024. Hasilnya ditunjukan pada Tabel 7.

Tabel 7. Hasil pengujian *dense layer*

Dense	*Avg Train Accuracy*	*Avg Train Loss*	*Avg Val Accuracy*	*Avg Val Loss*
256	**99.12%**	**0.02**	**94.20%**	**0.39**
512	99.35%	0.02	95.09%	0.68
1024	99.17%	0.01	95.09%	0.65

Hasil pengujian nilai *unit dense layer* menunjukan performa terbaik pada *dense layer* dengan nilai 256

5.1.2.5 Pengujian Jumlah layer dari Dense dan Dropout

Pengujian akan dilakukan dengan nilai *learning rate* 1e-4, *dropout* 0.5 dan *dense* 256. Jumlah *layer* yang akan diujikan adalah 1 dan 2 *dense layer* serta *dropout layer*. Hasilnya ditunjukan pada Tabel 8.

Tabel 8. Hasil pengujian *jumlah layer*

Layer	*Avg Train Accuracy*	*Avg Train Loss*	*Avg Val Accuracy*	*Avg Val Loss*
1	99.12%	0.02	94.20%	0.39
2	**99.21%**	**0.02**	**94.20%**	**0.37**

Hasil pengujian jumlah *layer* dari *dense* dan *dropout layer* menunjukan bahwa nilai *training loss* dan *validation loss* dari dua *layer* lebih optimal dibandingkan satu *layer*.

5.1.2.6 Pengujian Freeze Layer

Pengujian akan dimulai dengan melakukan *unfreeze* pada *layer* terakhir dari *pre-trained model ResNet50*. Proses pengujian akan dilakukan dengan membuka secara bertahap satu per satu *block* yang ada pada *layer conv5*. Proses pengujian akan dimulai dengan membuka *block* terakhir yaitu *conv5_block3* dan akan diikuti dengan *conv5_block2* dan *conv5_block1*. Pengujian ini akan dilakukan dengan 100 *epoch*. Hasilnya ditunjukan pada Tabel 9.

Tabel 9. Hasil pengujian *Freeze layer*

un-freeze Block	*Avg Train Accuracy*	*Avg Train Loss*	*Avg Val Accuracy*	*Avg Val Loss*
conv5_ block3	98.52%	0.04	93.30%	0.37
conv5_ block2	98.57%	0.04	93.75%	0.50
conv5_ block1	**99.12%**	**0.02**	**94.64%**	**0.33**

Hasil pengujian *unfreeze layer* menunjukan bahwa setiap melakukan *unfreeze layer* atau membuka *block* membuat peningkatan dari akurasi dan penurunan dari *loss* baik *train* maupun *validation*.

5.1.2.7 Pengujian Data Augmentation

Pengujian akan dilakukan dengan memanfaatkan percobaan sebelumnya yaitu *layer conv5_block1* yang di *unfreeze*. Proses pengujian dilakukan pada *train* tanpa *data augmentation* dan *train* dengan *data augmentation*. Hasilnya ditunjukan pada Tabel 10.

Tabel 10. Hasil pengujian *data augmentation*

Data Augmentation	*Avg Train Accuracy*	*Avg Train Loss*	*Avg Val Accuracy*	*Avg Val Loss*
Tanpa data augmentasi	99.17%	0.02	93.75%	1.87
Dengan data augmentasi	99.12%	0.02	94.20%	0.58

Hasil pengujian *data augmentation* menunjukan bahwa nilai *training loss* dan *validation loss* pada pengujian yang menggunakan *data augmentation* memperoleh nilai yang sangat optimal

5.1.2.8 Pengujian Akurasi, Presisi, Recall dan F-Score dari Model Fine-Tune

Proses pengujian ini dilakukan dengan semua konfigurasi yang telah digunakan sebelumnya yaitu *learning rate* 1e-4, jumlah *epoch* 100, 2 *layer dense* dengan nilai unit 256, *dropout layer* dengan nilai 0.5, *unfreeze layer* terakhir pada *pretrained model ResNet50 (conv5_block1, conv5_block2* dan *conv5_block3)* serta penggunaan *data augmentation*. Hasilnya ditunjukan pada Tabel 11.

Tabel 11. Hasil pengujian *performa dari fine-tune ResNet50*

Kategori	**Akurasi**	**Presisi**	***Recall***	***F-Score***
helmet	96%	0.96	0.96	0.96
no_helmet	93%	0.93	0.93	0.93
no_vest	94%	0.94	0.94	0.94
vest	98%	0.98	0.98	0.98

5.1.2.9 Perbandingan uji coba pada ResNet Sebelum dan Sesudah fine-tune

Pada bagian ini akan dilakukan perbandingan performa pada *model ResNet50* sebelum dan sesudah dilakukan *fine-tune* pada tiap kategori kelas. Performa yang akan dibandingkan adalah akurasi, presisi, *recall* dan *f-score* dari setiap *model*. Proses perbandingan ini akan dilakukan pada *epoch* 100. Hasilnya ditunjukan pada Tabel 12, Tabel 13, Tabel 14 dan Tabel 15.

Tabel 12. Perbandingan akurasi

Kategori	**Akurasi sebelum *fine-tune***	**Akurasi sesudah *fine-tune***
helmet	97%	96%
no_helmet	92%	93%
no_vest	91%	94%
vest	97%	98%

Tabel 13. Perbandingan presisi

Kategori	Presisi sebelum *fine-tune*	Presisi sesudah *fine-tune*
helmet	0.97	0.96
no_helmet	0.90	0.93
no_vest	0.90	0.94
vest	0.99	0.98

Tabel 14. Perbandingan *recall*

Kategori	*Recall* sebelum *fine-tune*	*Recall* sesudah *fine-tune*
helmet	0.96	0.96
no_helmet	0.94	0.93
no_vest	0.92	0.94
vest	0.96	0.98

Tabel 15. Perbandingan *f-score*

Kategori	*f-score* sebelum *fine-tune*	*f-score* sesudah *fine-tune*
helmet	97%	96%
no_helmet	92%	93%
no_vest	91%	94%
vest	97%	98%

Walaupun menunjukan performa yang tidak jauh berbeda, namun *model ResNet* sesudah *fine-tune* memiliki keunggulan sedikit lebih baik dibandingkan sebelum *fine tune*.

5.2 Pengujian Aplikasi

Pengujian Aplikasi dilakukan dengan melakukan deteksi dan klasifikasi berdasarkan *user interface* yang telah dibuat. Sistem aplikasi ini memerlukan waktu sekitar 3 detik untuk melakukan deteksi dan klasifikasi gambar yang diinputkan. Pengujian akan dilakukan pada empat kelas yang berbeda serta gambar yang tidak terdapat objek helm dan rompi. Contoh hasil deteksi dan klasifikasi yang ditampilkan oleh program ditunjukan pada Gambar 7 dan Gambar 8.

Gambar 7. Tampilan aplikasi saat melakukan deteksi

Gambar 8. Tampilan aplikasi saat melakukan klasifikasi

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengamatan dan pengujian terhadap sistem yang telah dibuat, dapat disimpulkan beberapa hal sebagai berikut:

- Hasil pengujian dengan metode YOLOv3 menunjukan akurasi yang cukup baik untuk setiap bagian tubuh yang dideteksi, dimana akurasi yang diperoleh untuk kepala adalah 64.09% sementara badan memperoleh nilai akurasi 63.03%.
- Hasil pengujian *recall* pada metode YOLOv3 menunjukan hasil yang baik untuk setiap bagian tubuh yang dideteksi, dimana recall yang diperoleh untuk kelas kepala adalah 88.22% dan untuk kelas badan memperoleh nilai 88.04%.
- Hasil pengujian performa *model ResNet50* yang telah dimodifikasi menunjukan hasil yang sangat baik dengan tingkat akurasi 96%.
- Proses deteksi menggunakan YOLOv3 memiliki kekurangan dimana tidak mampu mendeteksi semua objek kepala dan badan pada gambar yang diinputkan. Proses deteksi pada banyak objek hanya mampu berhasil mendeteksi 2 atau 3 objek saja.

6.2 Saran

Saran yang dapat diberikan untuk menyempurnakan dan mengembangkan sistem ini lebih lanjut antara lain:

- Menambah *dataset* yang bervariasi untuk *training* data pada tiap kategori alat pelindung diri, sehingga dapat lebih meningkatkan akurasi prediksi dari *model* yang dibangun.
- Memanfaatkan *pretrained model* lainnya untuk menemukan *model* dengan akurasi yang lebih baik.
- Melakukan pengujian *training dataset* dengan data augmentasi yang lebih bervariasi.
- Menambahkan klasifikasi untuk objek alat pelindung diri lainnya.

7. REFERENCES

[1] Annamraju, A. (2020, January 1). Helmet_Dataset. Diambil kembali dari Kaggle: https://www.kaggle.com/abhishek4273/helmet-dataset

[2] Ginanti, P. D. (2019, October). *Memahami Pentingnya Menggunakan Alat Pelindung Diri Saat Bekerja*. Diambil kembali dari Alodokter: https://www.alodokter.com/memahami-pentingnya-menggunakan-alat-pelindung-diri-saat-bekerja

[3] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR). doi:10.1109/CVPR.2016.90

[4] Li, J., Liu, H., Wang, T., Jiang, M., Wang, S., Li, K., & Zhao, X. (2017). Safety Helmet Wearing Detection Based on Image Processing and Machine Learning. International Conference on Advanced Computational Intelligence (ICACI). doi:10.1109/ICACI.2017.7974509

[5] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement.

[6] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788. doi:10.1109/CVPR.2016.91.

[7] Sakib, S., Ahmed, N., Kabir, A. J., & Ahmed, H. (2018). An Overview of Convolutional Neural Network: Its Architecture and Applications. *Preprints*. doi:10.20944/preprints201811.0546.v1

[8] Seong, H., Choi, H., Cho, H., Lee, S., Son, H., & Kim, C. (2017). Vision-Based Safety Vest Detection in a Construction Scene. International Symposium on Automation and Robotics in Construction. doi:10.22260/ISARC2017/0039

[9] Syin, J. (2019, December 4). *Hardhat and Safety Vest Image for Object Detection*. Diambil kembali dari kaggle: https://www.kaggle.com/johnsyin97/hardhat-and-safety-vest-image-for-object-detection

[10] Zhong, M., & Fei, M. (2019). A YOLOv3-based non-helmet-use detection for seafarer safety aboard merchant ships. *Journal of Physics: Conference Series*. doi:10.1088/1742-6596/1325/1/012096