

# Klasifikasi Helm Keselamatan Menggunakan Metode *Convolutional Neural Network* (CNN)

<sup>1</sup>Aldiana Nugroho, <sup>2</sup>Diah Arie Widhining K, <sup>3</sup>Farrady Alif Fiolana

<sup>1,2,3</sup> Program Studi Teknik Elektro, Universitas Islam Kadiri, Kediri

[aldiana.nugroho@gmail.com](mailto:aldiana.nugroho@gmail.com), [diahariewk@uniska-kediri.ac.id](mailto:diahariewk@uniska-kediri.ac.id), [farradyalif@uniska-kediri.ac.id](mailto:farradyalif@uniska-kediri.ac.id)

**Abstract** - The industrial world that uses human resources is fully related to the production process which requires the application of the latest technology. The ability of technology to improve production processes can also be dangerous when combined with unsafe working conditions, procedures and systems. In addition, both employers and employees usually ignore the presence of hazards to achieve production goals. Occupational Safety and Health (K3) has been implemented in every business to create a safe work environment, work processes and work systems. So an early stage system was created that could later differentiate workers who wore helmets or not, which would make it easier for workers as a reminder to wear safety helmets. Using the CNN Architecture to classify the image of workers wearing helmets or not, the results of the study are first to analyze the results of the convolution on one of the images and display all the kernels used in the convolution process to produce results, namely the output image provides a fairly clear visual with the initial image, second CNN Architecture Evaluation using Confusion Matrix gives accuracy results of 81% and Mean Square Error of 18.8%. The last test is testing the predictions of other images.

**Keywords** — Safety helmets, classification, cnn

**Abstrak**— Dunia industri yang menggunakan sumber daya manusia, sepenuhnya terkait dengan proses produksi yang membutuhkan penerapan teknologi mutakhir. Kemampuan teknologi untuk meningkatkan proses produksi juga bisa berbahaya jika digabungkan dengan kondisi, prosedur, dan sistem kerja yang tidak aman. Selain itu, baik pengusaha maupun karyawan biasanya mengabaikan adanya bahaya untuk mencapai tujuan produksi. Keselamatan dan Kesehatan Kerja (K3) telah diterapkan di setiap bisnis untuk menciptakan lingkungan kerja yang aman, proses kerja, dan sistem kerja. Sehingga dibuatkanlah sistem tahap awal yang nantinya bisa membedakan pekerja yang memakai helm atau tidak, yang nantinya bisa memudahkan para pekerja sebagai pengingat untuk memakai helm keselamatan. Menggunakan Arsitektur CNN untuk mengklasifikasi citra pekerja yang memakai helm atau tidak, memberikan hasil dari penelitian yaitu pertama menganalisa hasil konvolusi pada salah satu citra dan menampilkan seluruh kernel yang digunakan pada proses konvolusi membuah hasil yaitu citra output memberikan visual yang cukup jelas dengan citra awal, kedua yaitu Evaluasi Arsitektur CNN dengan menggunakan Confusion Matrix memberikan hasil accuracy sebesar 81% dan Mean Square Error sebesar 18,8%. Pengujian Terakhir yaitu menguji prediksi gambar lainnya.

**Kata Kunci**— Helm keselamatan, klasifikasi, cnn

## I. Pendahuluan

Dunia industri yang menggunakan sumber daya manusia, sepenuhnya terkait dengan proses produksi yang membutuhkan penerapan teknologi mutakhir. Kemampuan teknologi untuk meningkatkan proses produksi juga bisa berbahaya jika digabungkan dengan kondisi, prosedur, dan sistem kerja yang tidak aman. Selain itu, baik pengusaha maupun karyawan biasanya mengabaikan adanya bahaya untuk mencapai tujuan produksi. Keselamatan dan Kesehatan Kerja (K3) telah diterapkan di setiap bisnis untuk menciptakan lingkungan kerja yang aman, proses kerja, dan sistem kerja. Sehingga dibuatkanlah sistem tahap awal yang nantinya bisa membedakan pekerja yang memakai helm atau tidak, yang nantinya bisa memudahkan para pekerja sebagai pengingat untuk memakai helm keselamatan.[1]

Semua pihak yang terlibat dalam bisnis harus mempertimbangkan keselamatan dan kesehatan kerja. Hal tersebut diatur dalam UU Ketenagakerjaan No. 13 Tahun 2003 pada pasal 86 dan 87 mengenai Keselamatan dan Kesehatan Kerja (K3). Tujuan dasar dari kedua pasal ini adalah untuk memberikan perlindungan bagi pekerja untuk keselamatan dan kesehatan kerja melalui penerapan sistem manajemen keselamatan dan kesehatan kerja dalam bisnis. Sistem tersebut diterapkan untuk mengontrol dan menyediakan fasilitas kesehatan, serta untuk menghindari kecelakaan dan penyakit akibat kerja. untuk tempat kerja yang nyaman, efektif, dan produktif.[2]

Pada tahun 2022 Rescky Marthen Mailoa dan teman-temannya berhasil melakukan deteksi rompi dan helm keselamatan menggunakan metode YOLO dan CNN. *You Only Look Once* (YOLO) digunakan untuk mendeteksi bagian tubuh kepala dan badan, sementara itu *Convolutional Neural Network* (CNN) digunakan untuk memproses bagian tubuh yang terdeteksi. Hasil pengujian menunjukkan *accuracy* untuk kepala adalah 64,09% sementara badan memperoleh nilai *accuracy* sebesar 63,03%.[3]

Pada penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) sebagai langkah awal untuk membuat sistem dalam pengawasan disiplin Keselamatan dan Kesehatan Kerja (K3) sehingga resiko kecelakaan kerja dapat berkurang.

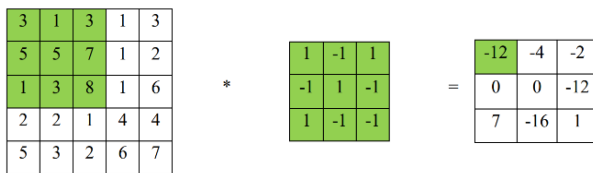
### A. Convolutional Neural Network

Convolutional Neural Network dikategorikan dalam deep learning karena aplikasinya yang luas dalam memproses data gambar dan kedalaman jaringannya yang tinggi. Convolutional Neural Network memiliki desain beberapa arsitektur yang dapat dipelajari. Feature map adalah keluaran yang digunakan sebagai input dan output dari setiap langkah.[4]

### B. Konsep CNN

#### 1. Convolutional Layer

sebagai lapisan pertama pada arsitektur jaringan yang langsung menerima masukan dari gambar adalah lapisan *convolutional*. Tujuan utama dari konvolusi data gambar adalah untuk mengekstrak fitur dari data gambar yang masuk.[5]



Gambar 1.1 Convolutional Layer

Pada contoh gambar diatas menggunakan *input*  $5 \times 5$  dan kernel dengan ukuran  $3 \times 3$  yang dimulai dari sisi kiri, prosedur ini dikenal sebagai *sliding window*. Ilustrasi cara menghitung proses *convolution* adalah sebagai berikut.

$$a \cdot b = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n \quad (1.1)$$

$$\text{Posisi 1} = (3 \times 1) + (1 \times (-1)) + (3 \times 1) + (5 \times (-1)) + (5 \times 1) + (7 \times (-1)) + (1 \times 1) + (3 \times (-1)) + (8 \times (-1)) = -12$$

$$\text{Posisi 2} = (1 \times 1) + (3 \times (-1)) + (1 \times 1) + (5 \times (-1)) + (7 \times 1) + (1 \times (-1)) + (3 \times 1) + (8 \times (-1)) + ((-1) \times (-1)) = -4$$

untuk lebih memahami cara kerja *convolution*. Persamaan berikut dapat digunakan untuk menentukan *feature map*.

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1 \quad (1.2)$$

Keterangan:

- $n_{in}$  : number of input features
- $n_{out}$  : number of output features
- $k$  : convolution kernel size
- $p$  : convolution padding size
- $s$  : convolution stride size

Terdapat dua parameter untuk memodifikasi layer, yaitu:

- a. *Stride* adalah parameter yang berfungsi untuk mengontrol jumlah pergeseran kernel, jika nilai *stride* satu langkah, *feature map* akan bergeser satu pixel baik secara *horizontal* maupun *vertikal*. [6]
- b. *Padding* adalah parameter yang berfungsi untuk menentukan jumlah *pixel* (bernilai nol) yang ditambah pada tiap sisi dari *input*. *Padding* digunakan untuk membuat nilai *output* sama dengan nilai input. [7]

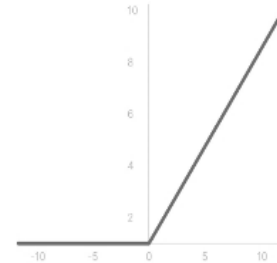
#### 2. Activation Function

*Activation function* adalah sebuah *node* yang ditambahkan di akhir keluaran dari setiap jaringan syaraf. Fungsi aktivasi dapat diterapkan baik selama atau setelah prosedur lapisan konvolusi dalam desain *Convolutional Neural Network*. Dalam jaringan saraf berbagai fungsi aktivasi sering digunakan, antara lain: [8]

##### a. ReLU (Rectify Linear)

Aktivasi *ReLU (Rectify Linear)* diaktifkan menggunakan karakteristiknya bahwa jika *input negatif*, *output neuron* dapat direpresentasikan sebagai 0. *Output* dari *neuron* adalah nilai *input* dari aktivasi itu sendiri jika nilai *input* dari fungsi aktivasi positif.

$$n_{out} = \left[ \frac{n_{in} + 2p - k}{s} \right] + 1 \quad (1.3)$$

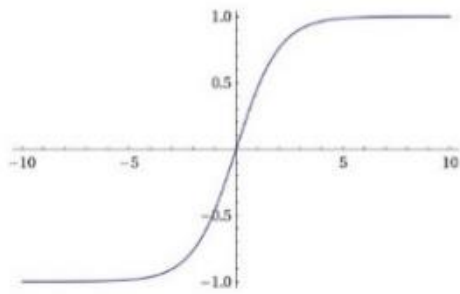


Gambar 1.2 Fungsi aktivasi ReLU

##### b. Softmax

Fungsi *softmax* adalah fungsi aktivasi lain yang cukup signifikan untuk digunakan pada akhir setiap model. Fungsi utama *Softmax* dalam proses klasifikasi adalah untuk menentukan kemungkinan setiap target kelas dibandingkan dengan setiap target kelas lainnya. Jika semua probabilitas untuk kelas target digabungkan bersama, rentang probabilitas output untuk *softmax* adalah angka antara 0 dan 1, dan akan sama dengan 1.

$$\text{softmax}(Z_i) = \frac{\exp(Z_i)}{\sum_j \exp(Z_j)} \quad (1.4)$$

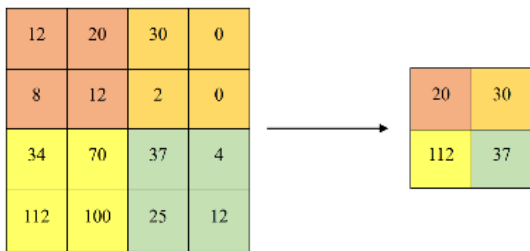


Gambar 1.3 Fungsi softmax

3. Pooling Layer

Pooling adalah pengurangan ukuran matriks sering dilakukan setelah operasi convolutional layer.[9] Pada umumnya pendekatan yang sering digunakan adalah max pooling, atau mengambil nilai terbesar pada bagian tersebut.

$$Pool_{x,y} = Max(Conv_{x,y}, Conv_{x+1,y}, Conv_{x,y+1}, Conv_{x+1,y+1}) \quad (1.5)$$

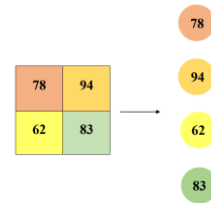


Gambar 1.4 Max Pooling

Pooling digunakan berujuan untuk mempercepat proses perhitungan. Hal ini dimungkinkan karena lebih sedikit parameter yang perlu diubah setelah melewati pooling layer, sehingga bahaya overfitting berkurang. Pooling memiliki stride yang lebih panjang dan output yang lebih kecil. Hal ini cukup bermanfaat dalam membantu inisiatif efisiensi tersebut.

4. Flatten

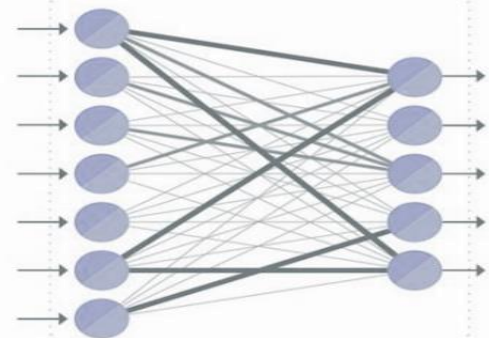
Tujuan dari flatten adalah untuk mengubah multidimensional array yang merupakan feature map menjadi vektor. Ini diperlukan agar fully connected sepenuhnya dapat menggunakan data ini sebagai input. Oleh karena itu, diperlukan suatu fungsi yang dapat mengubah data multidimensional array ke dalam bentuk vektor.[10]



Gambar 1.5 Flatten

5. Fully Connected Layer

Fully Connected Layer merupakan feed forward neural network yang terdiri dari hidden layer, activation function, output layer, dan loss function. Fully Connected Layer sering digunakan untuk memodifikasi parameter data sehingga data dapat dikategorikan dengan benar. Layer ini menerima data dalam bentuk vektor sebagai masukan dari proses feature learning.



Gambar 1.6 Fully Connected Layer

6. Confusion Matrix

Nilai accuracy model merupakan salah satu cara untuk mengukur kinerja model klasifikasi. True positives (TP), True Negatives (TN), False Positives (FP), dan False Negatives (FN) adalah beberapa parameter yang berfungsi sebagai dasar untuk pencarian nilai yang akurat. Confusion Matrix yaitu matriks yang sering merangkum parameter tersebut, terlihat pada gambar berikut:

		Actual Class	
		1	0
Predicted Class	1	True Positive	False Positive
	0	False Negative	True Negative

Gambar 1.7 Confusion Matrix

Terdapat istilah pada representasi hasil dari *confusion matrix*, [11] yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN), berikut penjelasan untuk memahami:

a. *True Positive* (TP)

Merupakan hasil positif yang diprediksi benar, contohnya ground truth atau input menyatakan bahwa citra tersebut helm namun prediksi atau algoritma menyatakan bahwa citra tersebut helm

b. *False Positive* (FP)

Merupakan data negative namun diprediksi sebagai data positive, contohnya ground truth atau input menyatakan bahwa citra tersebut non\_helm namun prediksi atau algoritma menyatakan bahwa citra tersebut helm

c. *True Negative* (TN)

Merupakan data negative yang diprediksi benar, contohnya ground truth atau input menyatakan bahwa citra tersebut non\_helm namun prediksi atau algoritma menyatakan bahwa citra tersebut non\_helm

d. *False Negative* (FN)

Merupakan data positive yang diprediksi sebagai data negative, contohnya ground truth atau input menyatakan bahwa citra tersebut helm namun prediksi atau algoritma menyatakan bahwa citra tersebut.

*Confusion matrix* digunakan dalam model klasifikasi biner yang hanya membedakan antara dua jenis kelas. *Confusion matrix* memberikan informasi tentang kesalahan model dan mengilustrasikan bagaimana model memprediksi kelas, di antara kelebihan lainnya.

7. *Accuracy*

*Accuracy* menggambarkan seberapa baik model mengklasifikasikan kumpulan data. Se jauh mana nilai yang diproyeksikan dekat dengan nilai sebenarnya adalah *accuracy*.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1.6)$$

		1	0
Predicted Class	1	True Positive	False Positive
	0	False Negative	True Negative

Gambar 1.8 *Accuracy*

8. *Precision*

*Precision* menggambarkan rasio prakiraan positif asli terhadap jumlah total prediksi yang dibuat, di mana akurasi adalah tingkat kesesuaian antara data yang diinginkan dan hasil yang diproyeksikan oleh model yang dibuat.

$$Precision = \frac{TP}{TP + FP} \quad (1.7)$$

		1	0
Predicted Class	1	True Positive	False Positive
	0	False Negative	True Negative

Gambar 1.9 *Precision*

9. *Recall*

*Recall* atau *Sensitivity* (*True Positive Rate*) menggambarkan persentase prediksi akurat untuk semua data akurat menunjukkan seberapa baik model tersebut memperoleh informasi.[12]

$$Recall = \frac{TP}{TP + FN} \quad (1.8)$$

		1	0
Predicted Class	1	True Positive	False Positive
	0	False Negative	True Negative

Gambar 1.10 *Recall*

10. *F1-Score*

*F1-Score* menggambarkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan.[13]

$$F1 - Score = \frac{(2 \times Recall \times Precision)}{(Recall + Precision)} \quad (1.9)$$

11. *Mean Square Error*

Fungsi loss *Mean Squared Error* (MSE) adalah jumlah selisih kuadrat antara entri dalam vektor prediksi  $y$  dan vektor *ground truth*  $y_{hat}$ . Fungsi *loss* ini digunakan

untuk mengetahui seberapa dekat garis regresi dengan sekumpulan titik data. Fungsi *loss* ini juga dikenal sebagai *L2 regularization*

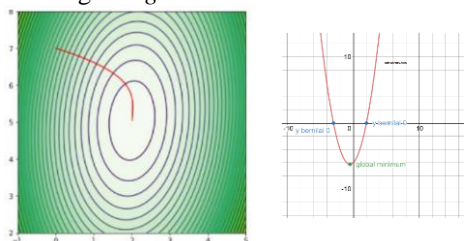
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1.10)$$

### 12. Batch Size

*Batch Size* dalam *machine learning* dan *deep learning* mengacu pada volume data yang digunakan selama satu siklus komputasi. Pendekatan yang paling efektif akan selalu diadopsi, bahkan ketika ada sejumlah metode yang memiliki kelebihan dan kekurangan. Konsep metode antara lain:

#### a. Gradient Descent

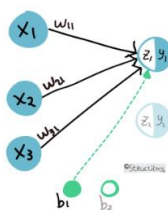
*Gradient Descent* adalah algoritma optimasi yang sering digunakan untuk menemukan *weight* atau koefisien algoritma *machine learning*. Gambar dibawah ini merupakan ilustrasi dari *gradient descent* yang menggunakan seluruh data *training* dalam sekali proses komputasi. Tentu dalam hal ini sangat berbahaya karena sangat rentan sekali untuk terjebak dalam *saddle point*, karena hanya mengikuti *gradient loss* dari data latih tersebut.[14]



Gambar 1.11 Gradient Descent

### 13. Forward Propagation

*Forward propagation* adalah metode yang digunakan untuk mengirimkan data dari *input layer* melalui setiap *neuron hidden layer* ke *output layer*, dimana *error* akan ditentukan dengan menggunakan rumus yang dikenal dengan *loss function*. [15] *Loss Function* digunakan untuk mengukur seberapa baik kinerja *deep neural network*.



Gambar 1.12 Forward Propagation

Untuk perhitungannya, nilai  $y_1$  diperoleh dengan menghitung nilai  $z_1$  terlebih dahulu (perhitungan linier).

$$z_1 = w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + b_1 \quad (1.11)$$

Setelah diperoleh  $z_1$ , output prediksi  $y_1$  diperoleh dengan menerapkan fungsi aktivasi terhadap  $z_1$ .

$$y_1 = \sigma(z_1) \quad (1.12)$$

Perhitungan untuk semua  $y$  secara umum bisa menggunakan rumus:

$$y_j = \sigma \left( \sum_{i=1}^N w_{ij}x_i + b_j \right) \quad (1.13)$$

### 14. Back Propagation

Bobot *neural network* ditentukan melalui *back propagation* menggunakan teknik seperti *gradient descent*. Karena komputasi dimulai dari *output jaringan saraf ke input*, prosedur ini dikenal sebagai *back propagation*.

Untuk memperbaiki suatu bobot  $w$  berdasarkan *error E* yaitu:

$$w_{new} = w_{old} - \alpha \frac{\partial E}{\partial w} \quad (1.14)$$

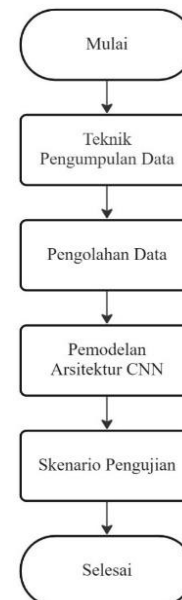
Untuk memperbaiki nilai bias:

$$b_{new} = b_{old} - \alpha \frac{\partial E}{\partial b} \quad (1.15)$$

Simbol  $\alpha$  pada persamaan di atas adalah *learning rate*.

## II. Metode Penelitian

### A. Metode Penelitian



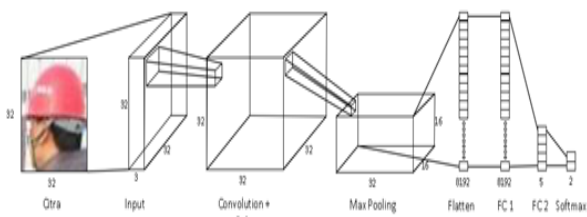
Gambar 2.1 Flowchart Penelitian

1. Teknik Pengumpulan Data  
Pada alur ini metode penelitian yang pertama yaitu berisi tentang bagaimana dataset helm keselamatan diperoleh dan melakukan pengecekan isi dari dataset citra helm keselamatan
2. Pengolahan Data

Pada alur kedua dari metode penelitian yang kedua yaitu berisi tentang bagaimana dataset citra helm keselamatan diolah sebelum diproses.

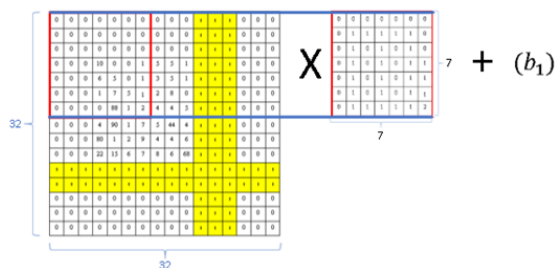
3. **Arsitektur Convolutional Neural Network (CNN)**  
 Pada alur dari metode penelitian yang ketiga yaitu berisi tentang memodelkan arsitektur dari deep learning yang digunakan pada penelitian ini digambarkan dan dijelaskan secara sederhana dan dapat dipahami.
4. **Skenario Pengujian**  
 Pada alur dari metode penelitian yang terakhir berisi tentang tahapan-tahapan pengujian yang dilakukan pada penelitian ini.

**B. Arsitektur Convolutional Neural Network**



Gambar 2.2 Model Arsitektur Convolutional Neural Network

1. **Input**  
 Langkah pertama yaitu input sebuah citra yang nantinya digunakan untuk pemrosesan kedalam arsitektur Convolutional Neural Network. Citra yang masuk berukuran 32x32 pixel dengan 3 layer RGB. Data input sudah diolah terlebih dahulu pada Image Preprocessing, data telah dirubah kedalam bentuk data array RGB, sehingga data telah siap untuk masuk kedalam proses kovolusi.
2. **Proses Convolution dan ReLU**  
 Pada langkah Selanjutnya bagian convolution + ReLU, citra yang masuk sebagai input dengan ukuran sebesar 32x32 pixel dan 3 layer RGB (Red, Green dan Blue).



Gambar 2.3 Ilustrasi proses Convolution pada arsitektur

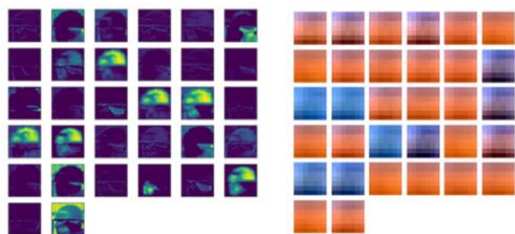
Pada gambar diatas digambarkan 1 layer dari citra yang melalui proses convoltion. Memberikan zero padding sebanyak 3 pada masing-masing layer citra. Melakukan proses convolution dengan scanning kernel

pada tiap layer dengan kernel ukuran 7x7 pixel. dikarenakan dalam 1 halaman tidak cukup untuk menggambarkan 32 pixel, sehingga diberikan tanda penjelas (blok warna kuning) bahwa layer tersebut 32 pixel

3. **MaxPooling**  
 Pada langkah Selanjutnya bagian MaxPooling, hasil output dari proses convolution + ReLU akan masuk sebagai input dengan ukuran sebesar 32x32 pixel dan memiliki ketebalan 32. Tahapan MaxPooling menggunakan ukuran kernel 2x2 dan stride 2 langkah. Sehingga nilai output dari Maxpooling yaitu (N, 32, 16, 16), N berasal dari jumlah data train dan validation
4. **Proses Flatten**  
 Setelah didapatkan nilai output dari proses MaxPooling, langkah Selanjutnya yaitu Flatten, nilai Flatteng didapatkan dengan perhitungan matrix (32, 16, 16) menjadi matrix 1 Dimensi, total yang didapatkan yaitu 32 x 16 x 16 = 8192 jumlah data yang di-flatten. Sehingga Output dari Flatten yaitu (N, 8192)
5. **Proses Fully Connected 1**  
 Pada proses fully connected 1 nilai masukan yang didapatkan dari proses fatten yaitu (N, 8192) . Selanjutnya melakukan proses perkalian antara masing-masing nilai yang didapatkan dari nilai output flatten. Sehingga nantinya akan menghasilkan output yang digunakan untuk proses Fully Connected 2. Setelah didapatkan hasil Output yaitu proses fully connected 1, proses selanjutnya proses Relu, yaitu mengeliminasi nilai-nilai yang bernilai negatif dengan menjadi nilai 0 dan tetap mempertahankan nilai positif.
6. **Proses Fully Connected 2**  
 selanjutnya melakukan proses perkalian antara masing-masing nilai yang didapatkan dari nilai output fully connected 1. Sehingga akan menghasilkan output yang digunakan untuk proses output yaitu mentukan hasil kelas mana dalam proses klasifikasi.

**III. Hasil dan Pembahasan**

Setiap model dalam penelitian ini yang telah menjalani training menghasilkan angka accuracy dan loss untuk setiap prosesnya. Model terbaik kemudian dipilih dengan membandingkan hasil tes. Waktu pelatihan, akurasi pelatihan, kerugian pelatihan, dan akurasi pengujian adalah beberapa karakteristik yang signifikan dalam memilih model untuk melanjutkan ke tahap evaluasi.



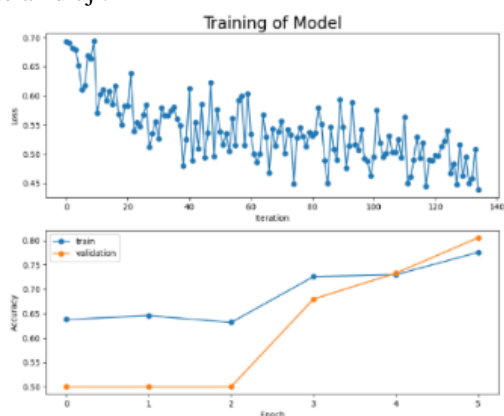
Gambar 3.1 Hasil convolution dan kernel

Pada gambar bagian kanan adalah kumpulan filter yang berjumlah 32 dengan ukuran 7x7 yang nantinya digunakan untuk proses konvolusi. Filter yang berjumlah 32 dan memiliki ukuran 7x7 pixel akan melalui proses perkalian dengan nilai pixel input gambar. Pada gambar diatas, dapat kita ketahui pada proses konvolusi pada iterasi pertama tentu cukup.

Setelah arsitektur dibentuk, selanjutnya akan dilakukan pelatihan langsung terhadap data yang sudah tersedia. Mencari nilai bobot dengan nilai *loss* 18,8% dan *accuracy* 81% merupakan tujuan dari pelatihan ini. Sekitar 70% data akan digunakan untuk *training*, dan 20% untuk *validation*. Untuk menghasilkan bobot pembelajaran, algoritma CNN akan menggunakan data tersebut.

Pengaturan iterasi pada proses pelatihan dan validasi membutuhkan 5 *epoch* dan *batch size* 128. Untuk mendapatkan bobot pembelajaran terbaik, prosedur *training* dan *validation* dilakukan sebanyak lima kali. Oleh karena itu, 0,001 adalah nilai *learning rate* yang diterapkan selama prosedur *training* dan *validation*.

Gambar dibawah ini menampilkan grafik *accuracy* dan *loss* selama *training* model dengan menggunakan 5 *epoch*. Fase *validation*, yang melihat proses masing-masing *epoch* pada model, juga ditunjukkan bersama dengan proses *training*. Fase *validation*, yang mengevaluasi pada setiap proses *epoch* model, juga ditunjukkan bersama dengan proses *training*. Tidak ada perbedaan yang cukup jauh antara temuan *training* dan *validation*. Ini menunjukkan seberapa sukses model yang dibuat telah diuji.

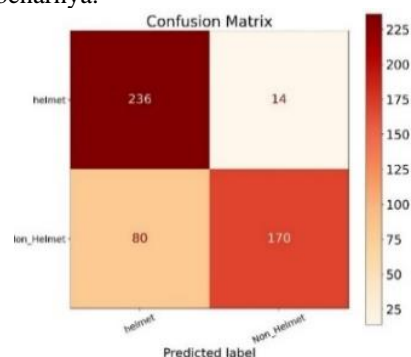


Gambar 3.2 Grafik accuracy dan loss

Setelah model dilatih, fase evaluasi adalah yang terakhir. Evaluasi ini akan mengukur seberapa baik performa model klasifikasi helm pengaman. Menghitung nilai *precision*, *recall*, *f1-score*, dan *accuracy* model adalah salah satu cara untuk menilai seberapa baik kinerjanya sebagai model klasifikasi. Menggunakan parameter *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) adalah beberapa terminologi yang sering digunakan dalam menentukan performa nilai *precision*, *recall*, *f1-score*, dan *accuracy*.

Dalam CNN, pembuatan model yang siap pakai untuk digunakan analisis disebut juga evaluasi merupakan hasil akhir dari proses pelatihan atau pelatihan. Ada dua kriteria evaluasi untuk setiap proses pelatihan yaitu *accuracy* dan *loss*. Nilai *loss* adalah angka yang menyatakan seberapa tidak akurat perkiraan model untuk sampel data. Nilai *loss* akan menjadi 0 jika perkiraan model akurat. Sementara *accuracy* penting ketika model membuat prediksi yang akurat.

Untuk mengukur seberapa baik setiap model bekerja, *confusion matrix* akan menghasilkan sejumlah nilai. *Precision*, *recall*, *f1-score*, dan *accuracy* adalah hasil akhirnya. Variabel-variabel ini ditentukan dengan menggunakan komponen *confusion matrix*, yang mewakili jumlah hasil dibandingkan dengan kelas prediksi sebenarnya.



Gambar 3.3 Confusion Matrix

Berdasarkan hasil pengujian dan analisis, dapat dikatakan bahwa metode *Convolutional Neural Network* (CNN) dapat dianggap efektif untuk mengkategorikan helm pengaman. Kedua model dengan arsitektur lapisan konvolusi dan lima lapisan *fully connected* sepenuhnya menunjukkan kinerja klasifikasi yang baik, dengan skor rata-rata lebih tinggi dalam *f1-score*.

Tabel 3.1 Hasil pengujian Confusion Matrix

Model	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Accuracy</i>	<i>MSE</i>
Helmet	0.75	0.94	0.83	0.81	0.188
Non-Helmet	0.92	0.68	0.78	0.81	

*F1-score* 83%, *recall* 94%, *precision* 75%, dan tingkat *accuracy* 81%. Teknik *Convolutional Neural Network* (CNN) memiliki tingkat *accuracy* 81%, yang menunjukkan bahwa

teknik ini bekerja cukup baik dalam studi tentang klasifikasi helm pengaman.



Gambar 3.4 Hasil prediksi benar

Gambar diatas merupakan salah satu dari beberapa hasil prediksi yang ditampilkan. Model yang dikerjakan mendapatkan nilai *accuracy* yang tinggi/mendekati sempurna, terdapat citra helm keselamatan yang salah diprediksi oleh model *Convolutional Neural Network*. Banyak penyebab untuk memastikan mengapa model ini mengalami kesalahan dalam memprediksi citra, salah satunya dikarenakan pada model pembelajaran gambar yang diuji identik dengan prediksi yang salah, ukuran *pixel* yang berbeda dan dataset gambar yang digunakan sebagai image detection yang digunakan sebagai gambar pada dataset untuk klasifikasi. Model dari *Convolutional Neural Network* mengira bahwa citra orang yang menggunakan object selain helm keselamatan seperti songkok di klasifikasikan lebih mendekati ke kelas *Helmet*, dikarenakan memang bentuk dan warna hampir mendekati. Prediksi model *Convolutional Neural Network* diuji terhadap kumpulan data luar dari data gambar helm dan non-helm.

#### IV. Kesimpulan

Berdasarkan pada proses pengujian, pengamatan dan analisis hasil yang telah dilakukan dapat disimpulkan bahwa:

1. Arsitektur CNN dalam menyelesaikan permasalahan klasifikasi objek gambar berupa helm keselamatan yang dipakai pekerja industri. Pada tahapan-tahapan arsitektur CNN berhasil menampilkan hasil dari kernel yang digunakan pada proses *convolution* dan *feature map* yang merupakan hasil dari proses *convolution*. Masing-masing jumlah kernel berjumlah 32 buah dan berukuran masing kernel 7x7 pixel. Pada hasil *feature map*, menghasilkan 32 buah *feature map* pada hasil dari proses *convolution*.
2. Pada pengujian hasil prediksi dari klasifikasi pada objek helm keselamatan dan non-helm mendapatkan hasil evaluasi menggunakan *Confusion Matrix* dengan tingkat presentase prediksi benar sebesar 81%. Tingkat presentasi keberhasilan prediksi didapatkan dari nilai prediksi benar pada kelas helm sejumlah 236 citra dan 170 citra untuk kategori *non-helmet*. Tentunya masih terdapat beberapa kesalahan dalam memprediksi, Melalui hasil evaluasi menggunakan *Mean Squared Error* didapatkan

presentase sebesar 18,8%. Sistem yang gagal memprediksi citra helm sejumlah 80 dan citra non-helm sejumlah 14 sehingga didapatkan nilai MSE sebesar 18,8%. Berdasarkan kesalahan yang dikarenakan pada model pembelajaran gambar yang diuji identik dengan prediksi yang salah, ukuran *pixel* yang berbeda dan dataset gambar yang digunakan sebagai citra deteksi yang digunakan sebagai gambar pada dataset untuk klasifikasi.

Saran yang dapat diberikan untuk pengembangan penelitian ini di masa mendatang adalah sebagai berikut:

1. Menggunakan gambar yang lebih jelas, tentunya dengan menggunakan gambar yang memiliki pixel yang lebih tinggi.
2. Menggunakan arsitektur *cnn* yang lebih bagus dan efektif, seperti menggunakan *hyperparameter adaptive learning rate, size image, batch size* yang lebih bervariasi.
3. Untuk penelitian selanjutnya, dari presentase kesalahan diharapkan lebih diperhatikan dalam membagi data setiap labelnya agar tidak salah saat memprediksi citra.

#### V. Daftar Pustaka

- [1] F. F. Rachman, H. Bethaningtyas, and R. F. Iskandar, "Analisis Sistem Deteksi Penggunaan Hard Hat Pada Pekerja Konstruksi Menggunakan Metode Jaringan Syaraf Tiruan Konvolusi," *eProceedings of Engineering*, vol. 8, no. 1, 2021.
- [2] B. Widodo, H. Armanto, and E. Setyati, "Deteksi Pemakaian Helm Proyek Dengan Metode Convolutional Neural Network," *INSYST: Journal of Intelligent System and Computation*, vol. 3, no. 1, pp. 23–29, 2021.
- [3] R. M. Mailoa and L. W. Santoso, "Deteksi Rompi dan Helm Keselamatan Menggunakan Metode YOLO dan CNN."
- [4] Y. Li, H. Wei, Z. Han, J. Huang, and W. Wang, "Deep Learning-Based Safety Helmet Detection in Engineering Management Based on Convolutional Neural Networks," *Advances in Civil Engineering*, vol. 2020, 2020, doi: 10.1155/2020/9703560.
- [5] M. Hendri, "Perancangan Sistem Deteksi Asap dan Api Menggunakan Pemrosesan Citra," 2018.
- [6] O. N. Putri, "Implementasi Metode Cnn Dalam Klasifikasi Gambar Jamur Pada Analisis Image Processing (Studi Kasus: Gambar Jamur Dengan Genus Agaricus Dan Amanita)," 2020.
- [7] A. Kamboj and N. Powar, "Safety Helmet Detection in Industrial Environment using Deep Learning," *Academy and Industry Research Collaboration Center (AIRCC)*, May 2020, pp. 197–208. doi: 10.5121/csit.2020.100518.



- 
- [8] M. R. Efrian and U. Latifa, "Image Recognition Berbasis Convolutional Neural Network (Cnn) Untuk Mendeteksi Penyakit Kulit Pada Manusia," *Power Elektronik: Jurnal Orang Elektro*, vol. 11, no. 2, pp. 276–282, 2022.
- [9] T. D. Antoko, M. A. Ridani, and A. E. Minarno, "Klasifikasi Buah Zaitun Menggunakan Convolution Neural Network," *Komputika: Jurnal Sistem Komputer*, vol. 10, no. 2, pp. 119–126, 2021.
- [10] R. S. Budi, R. Patmasari, and S. Saidah, "Klasifikasi Cuaca Menggunakan Metode Convolutional Neural Network," *eProceedings of Engineering*, vol. 8, no. 5, 2021.
- [11] D. T. Hermanto, A. Setyanto, and E. T. Luthfi, "Algoritma LSTM-CNN untuk Binary Klasifikasi dengan Word2vec pada Media Online," *Creative Information Technology Journal*, vol. 8, no. 1, pp. 64–77, 2021.
- [12] A. B. Prakosa, "Implementasi Model Deep Learning Convolutional Neural Network (Cnn) Pada Citra Penyakit Daun Jagung Untuk Klasifikasi Penyakit Tanaman," *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, vol. 6, no. 1, pp. 107–116, 2023.
- [13] E. Harytami, R. Y. Fajriatifah, and Y. H. Puspita, "Prediksi Penyakit Diabetes menggunakan Algoritma Artificial Neural Network," *Jurnal Data Science & Informatika*, vol. 2, no. 2, pp. 46–52, 2022.
- [14] N. D. Miranda, L. Novamizanti, and S. Rizal, "Convolutional Neural Network pada klasifikasi sidik jari menggunakan RESNET-50," *Jurnal Teknik Informatika (Jutif)*, vol. 1, no. 2, pp. 61–68, 2020.
- [15] M. F. Fadli, G. A. Buntoro, and F. Masykur, "Penerapan Algoritma Neural Network Pada Chatbot Pmb Universitas Muhammadiyah Ponorogo Berbasis Web," *JuSiTik: Jurnal Sistem dan Teknologi Informasi Komunikasi*, vol. 6, no. 1, pp. 13–22, 2022.