

CONVOLUTIONAL NEURAL NETWORK DALAM SISTEM DETEKSI HELM PADA PENGENDARA MOTOR

Ajib Susanto¹, Yupie Kusumawati², Ericsson Dhimas Niagara³, Christy Atika Sari⁴

^{1,3,4}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

²Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
e-mail: ¹ajib.susanto@dsn.dinus.ac.id, ²yupie@dsn.dinus.ac.id, ³111201811128@mhs.dinus.ac.id,
⁴christy.atika.sari@dsn.dinus.ac.id

ABSTRAK

Di tahun ini sudah terjadi perubahan atau revolusi dalam sistem industri yaitu revolusi industri 4.0, di mana industri sudah mengimplementasikan sebuah mesin-mesin yang serba cepat dan praktis untuk membantu dalam hal produktivitas. Dengan semakin pesatnya teknologi, banyak alat-alat yang di ciptakan untuk perkembangan teknologi di dunia. Pemanfaatan teknologi di gunakan untuk sebagai penggerak di pesatnya sistem berbasis AI ini dapat di dimanfaatkan untuk di berbagai bidang, salah satunya adalah penerapan sistem deep learning dan salah satu teknik yang terkenal dan sudah banyak digunakan untuk klasifikasi citra adalah menggunakan metode Jaringan Syaraf Tiruan (JST) yang mampu mengenali wajah manusia dan suatu gambar citra lalu mengklasifikasinya atau bisa di sebut image classification dan image recognition. Untuk perancangan sistem deteksi helm pada penelitian ini akan menggunakan algoritma CNN(Convolutional Neural Network). Convolutional Neural Network dan untuk model arsitektur yang di gunakan adalah MobileNetV2 dengan akurasi 80%, yang dapat di kategorikan baik dalam sebuah akurasi.

Kata Kunci: Revolusi Industri, Helm, Convolutional Neural Network, Deep learning, MobileNetV2

1. PENDAHULUAN

Di tahun ini sudah terjadi perubahan atau revolusi dalam sistem industri yaitu revolusi industri 4.0, di mana industri sudah mengimplementasikan sebuah mesin-mesin yang serba cepat dan praktis untuk membantu dalam hal produktivitas. Dengan semakin pesatnya teknologi, banyak alat-alat yang di ciptakan untuk perkembangan teknologi di dunia. Salah satu teknologi yang paling banyak yang di gunakan jaman sekarang adalah sistem berbasis kecerdasan buatan atau bisa di sebut AI (Artificial Intelligence) [1], [2]. Pemanfaatan teknologi di gunakan untuk sebagai penggerak di pesatnya sistem berbasis AI ini dapat di dimanfaatkan untuk di berbagai bidang, salah satunya adalah penerapan sistem deep learning dan salah satu teknik yang terkenal dan sudah banyak digunakan untuk klasifikasi citra adalah menggunakan metode Jaringan Syaraf Tiruan (JST) yang mampu mengenali wajah manusia dan suatu gambar citra lalu mengklasifikasinya atau bisa di sebut image classification dan image recognition.

Penelitian ini merupakan hasil survei prapenelitian yang dilakukan oleh peneliti terhadap 100 data konfirmasi surat E-tilang [3], 94% pengendara motor menggunakan helm. Di saat yang sama 6% tidak menggunakan helm dalam berkendara. Untuk perancangan sistem deteksi helm, dalam penelitian ini telah digunakan algoritma CNN (Convolutional Neural Network). Convolutional Neural Network [2], [4], [5] adalah salah satu algoritma pada deep learning yang dapat menerima sebuah inputan gambar dan di gunakan untuk mengidentifikasi dan mengenali sebuah gambar yang memiliki kelebihan yaitu mempunyai akurasi yang tinggi tetapi memiliki kelemahan yaitu proses pelatihan yang cukup lama. Dalam tahap awal proses pengembangan sistem deteksi helm ini, penulis melakukan percobaan perbandingan menggunakan beberapa arsitektur dari Convolutional Neural Network dan diharapkan dapat mendapatkan hasil yang maksimal. Dengan cara ini, keselamatan dalam berkendara akan lebih tinggi, dan dapat mengurangi korban jiwa saat terjadinya kecelakaan. Berdasarkan uraian latar belakang di atas, tahapan pembuatan sistem ini akan dibahas pada bagian selanjutnya untuk mengenal lebih lanjut tentang algoritma CNN.

2. TINJAUAN STUDI

Untuk meningkatkan kualitas sistem pelayanan masyarakat khususnya di Indonesia dalam menangani kasus pelanggaran tata tertib lalu lintas adalah dengan menerapkan suatu sistem yaitu e-Tilang. Dengan keberadaan sistem e-Tilang di Indonesia dapat membantu dalam mengurangi jumlah permasalahan yang terjadi pada proses penegakan pada hukum mengenai banyaknya pelanggaran tata tertib lalu lintas yang berlaku, di mulai dari aksi pungli, masalah sidang tilang dipengadilan hingga akutabilitas uang denda [3]. Pemantauan Internet of Thing (IoT) yang diterapkan antara lain pemantauan sensor gerak, pemantauan sensor ultrasonik, pemantauan sensor Passive Infra Red (PIR), dan pemantauan sensor kecepatan [6]. Metode pengenalan helm pengaman dari gambar resolusi rendah yang diambil dari video dan menyimpulkan hubungan antara fitur yang berbeda dan tingkat pengenalan dengan mempertimbangkan masalah pengenalan helm pengaman resolusi rendah pada jarak jauh [7], [8].

Kecelakaan lalu lintas merupakan salah satu penyebab utama kematian manusia. Di antara berbagai jenis kecelakaan di jalan, kecelakaan sepeda motor sering terjadi dan menyebabkan cedera parah. Helm merupakan pelindung utama pengendara sepeda motor. Sebagian besar negara mewajibkan penggunaan helm oleh pengendara sepeda motor, tetapi banyak orang gagal mematuhi hukum karena berbagai alasan. Kami menyajikan pengembangan sistem menggunakan pemrosesan gambar dan jaringan saraf konvolusi dalam (CNN) untuk menemukan pengendara sepeda motor yang melanggar undang-undang helm [9]. Sistem tersebut terdiri dari deteksi sepeda motor, klasifikasi helm vs tanpa helm, dan pengenalan plat nomor sepeda motor. Kami mengevaluasi sistem dalam hal akurasi dan kecepatan. Deteksi otomatis penggunaan helm sepeda motor melalui pengawasan video dapat memfasilitasi pendidikan yang efisien dan kampanye penegakan yang meningkatkan keselamatan di jalan [10], [11]. Namun, pendekatan deteksi yang ada memiliki sejumlah kekurangan, seperti ketidakmampuan untuk melacak sepeda motor individu melalui beberapa frame, atau untuk membedakan pengemudi dari penumpang dalam penggunaan helm. Selanjutnya, kumpulan data yang digunakan untuk mengembangkan pendekatan terbatas dalam hal lingkungan lalu lintas dan variasi kepadatan lalu lintas. Kami memperkenalkan metrik evaluasi untuk penggunaan helm dan akurasi deteksi pengendara, yang dapat digunakan sebagai tolok ukur untuk mengevaluasi pendekatan deteksi di masa mendatang. Tingginya angka kematian pada kecelakaan sepeda motor disebabkan oleh kelalaian pengendara roda dua dalam memakai helm. Identifikasi pengendara tanpa helm secara real-time adalah tugas penting untuk mencegah terjadinya peristiwa tersebut. Makalah ini telah menyajikan sistem otomatis untuk mengidentifikasi pengendara sepeda motor tanpa helm dari video pengawasan lalu lintas secara real-time. Dengan peningkatan bertahap teknologi kecerdasan buatan, pemrosesan gambar telah menjadi teknologi umum dan digunakan secara luas di berbagai bidang untuk menyediakan layanan berkualitas tinggi kepada orang-orang. Mulai dari algoritma computer vision dan teknologi pemrosesan gambar, sistem tampilan computer vision dirancang, dan algoritma koreksi distorsi gambar dieksplorasi untuk referensi [7], [12]. Kemajuan luar biasa telah dibuat dalam pengenalan gambar, terutama karena ketersediaan kumpulan data beranotasi skala besar dan jaringan saraf convolutional dalam (CNNs). CNN memungkinkan pembelajaran berdasarkan data, sangat representatif, fitur gambar hierarkis dari data pelatihan yang memadai. Namun, mendapatkan kumpulan data yang dijelaskan secara komprehensif seperti ImageNet dalam domain pencitraan medis tetap menjadi tantangan.

3. METODE PENELITIAN

3.1 CNN(Convolutional Neural Network)

CNN(Convolutional Neural Network) atau ConvNet adalah Algoritma *deep learning* dari *artificial neural network* yang mengadaptasi atau meniru cara kerja otak untuk membedakan dan mengklasifikasi sebuah gambar. Algoritma CNN di tujukan untuk data *input* berupa visual untuk, mengolah serta melatih *machine learning* untuk mengenali setiap objek visual yang telah di sediakan, semakin banyak *dataset* yang gunakan untuk melatih *machine learning* maka semakin akurat pada hasil *machine learning* [13]–[15]. Dibandingkan dengan algoritma klasifikasi lainnya, pra-pemrosesan yang diperlukan di ConvNet jauh lebih rendah. ConvNet sekarang menyediakan metode yang lebih terukur untuk klasifikasi gambar dan tugas pengenalan objek, menggunakan prinsip aljabar linier, terutama perkalian matriks, untuk mengenali pola dalam gambar. Dengan kata lain, mereka mungkin menuntut komputasi dan memerlukan unit pemrosesan grafis (GPU) untuk melatih model [4], [5], [16]–[18]. CNN mempunyai 3 lapisan utama, berikut adalah lapisan-lapisan tersebut:

1. Convolutional Layer (Lapisan konvolusi)

Lapisan konvolusi adalah blok bangunan inti CNN dan di mana sebagian besar perhitungan terjadi. Hal ini membutuhkan beberapa komponen yaitu *input data*, *filter* dan *feature map*. Mari kita asumsikan bahwa *inputnya* adalah gambar berwarna, yang terdiri dari matriks piksel 3D. Ini berarti bahwa *input* akan memiliki tiga dimensi yang tinggi, lebar, dan kedalaman yang sesuai dengan RGB pada gambar. CNN juga memiliki pendeteksi fitur, juga disebut kernel atau filter, yang akan bergerak pada bidang penerima gambar untuk memeriksa keberadaan fitur. Proses ini disebut konvolusi.

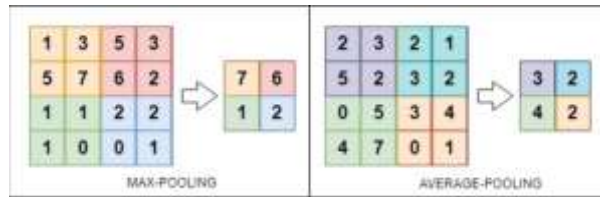
2. Pooling Layer (Lapisan pooling)

Lapisan *pooling*, juga disebut *downsampling*, melakukan pengurangan dimensi dan mengurangi jumlah parameter dalam *input*. Mirip dengan lapisan convolutional, operasi penyatuan filter melintasi *input*, tetapi perbedaannya adalah filter tidak berbobot. Sebagai gantinya, kernel menerapkan fungsi agregasi ke nilai di bidang reseptif, mengisi kolom keluaran. Lapisan pooling memiliki dua jenis utama yaitu *max pooling layer* dan *average max pooling*.

3. Fully-connected Layer

Fully-connected Layer adalah lapisan yang berfungsi untuk melakukan tugas klasifikasi berdasarkan fitur yang diekstraksi melalui lapisan sebelumnya dan filter yang berbeda. Sementara lapisan convolutional dan pooling

cenderung menggunakan fungsi ReLu, lapisan FC biasanya memanfaatkan fungsi aktivasi softmax untuk mengklasifikasikan input dengan tepat, menghasilkan probabilitas dari 0 hingga 1.



Gambar 1. Teknik Pooling

3.2 Image Dataset

Proses awal dalam dataset yang berupa kumpulan citra dan labelling, dimulai dengan menentukan label yang akan dibuat, dimulai dengan mengidentifikasi citra. Hasil dari tahap ini adalah labelling terhadap masing-masing citra. Masing-masih dataset dapat di lihat pada Gambar 2 dan 3.



Gambar 2. Dataset Citra Digital Helm



Gambar 3. Dataset Citra Digital Non_Helm

3.3 Classification Report

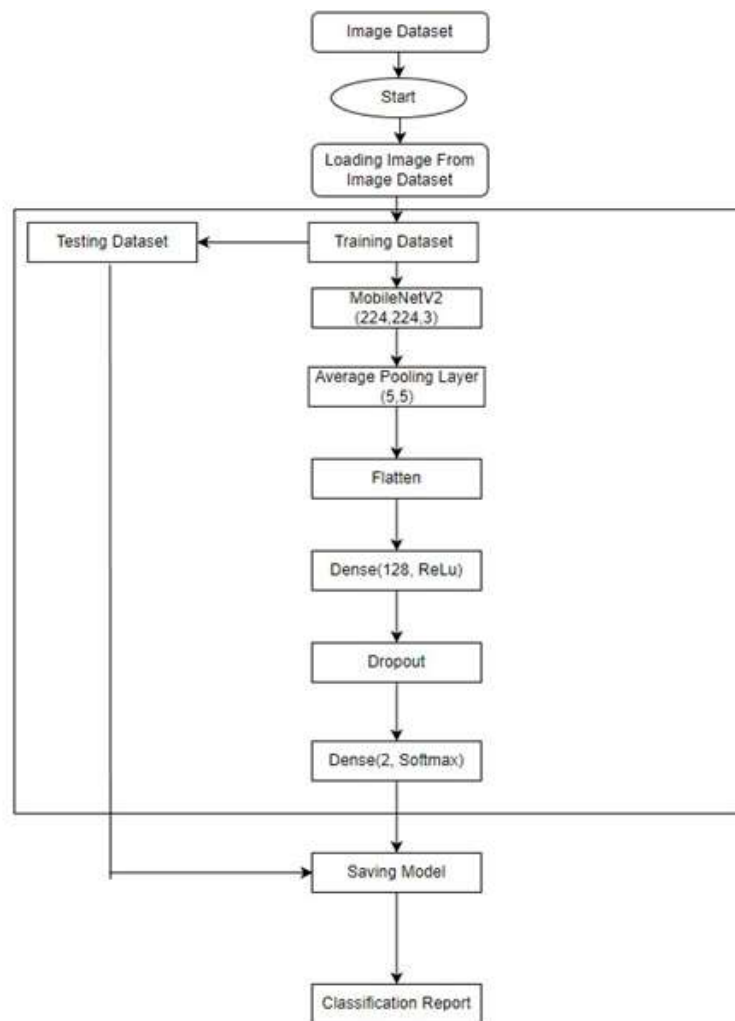
Tahap ini adalah tahap dalam memperlihatkan hasil dari semua proses di atas yang nantinya akan menghasilkan suatu evaluasi akurasi yang baik atau buruk dari model yang kita training. Untuk contoh classification report dapat di lihat pada Gambar 4.

	precision	recall	f1-score	support
helm	0.95	0.97	0.96	76
non_helm	0.97	0.95	0.96	75
accuracy			0.96	151
macro avg	0.96	0.96	0.96	151
weighted avg	0.96	0.96	0.96	151

Gambar 4. Classification Report

3.4 Usulan Metode

Dalam penelitian ini, perancangan pembangunan sistem deteksi helm menggunakan metode CNN (Convolutional Neural Network).



Gambar 5. Model Convolutional Neural Network

Berdasarkan Gambar 5, *MobileNetV2* berperan sebagai model dasar dari sistem deteksi ini. *MobileNetV2* akan menggunakan *input* 224,224,3 untuk melakukan klasifikasi pada *dataset* yang akan menjalankan *processing*. Proses *pooling layer* digunakan untuk mengurangi dimensi pada peta fitur. Oleh karena itu, proses ini mengurangi jumlah parameter yang akan dipelajari dan jumlah perhitungan yang dilakukan dalam jaringan. *Pooling layer* merangkum fitur-fitur yang ada di area peta fitur yang dihasilkan oleh lapisan konvolusi. Pada layer ini, terjadi

proses *flattening* yang merubah dari matriks yang ada di *pooling layer* menjadi satu kolom saja (sebuah vektor tunggal). Nantinya vektor ini akan menjadi bagian dari *input layer* di *convolutional neural networks* (CNN). *Dense Layer* adalah lapisan *neuron* sederhana di mana setiap *neuron* menerima *input* dari semua *neuron* di lapisan sebelumnya. *Dense Layer* digunakan untuk mengklasifikasikan gambar berdasarkan *output* dari *convolutional layer*. *Dropout* adalah teknik regularisasi jaringan syaraf dimana beberapa *neuron* akan dipilih secara acak dan tidak dipakai selama pelatihan. *Neuron-neuron* ini dapat dibuang secara acak. Proses berikutnya adalah menyimpan model yang sudah kita training untuk kita gunakan untuk sistem deteksi helm.

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Program

Pada hasil program ini adalah menampilkan hasil pendeteksian secara *real time* berupa *video stream*, dengan menyertakan *labelling* dan nilai akurasi pada *video stream* yang ada pada layar monitor. Hasil *output* memiliki dua *labelling* yaitu *helm* dan *no helm*.



Gambar 6. Tampilan deteksi melalui program

4.2 Pengujian

1. Pengujian BlackBox

Pada tahap pengujian melalui Behavioral (BlackBox) di perangkat lunak (software) dilakukan pada komponen antarmuka aplikasi / User Interface (UI). Pengujian ini dilakukan bertujuan untuk meminimalisir kesalahan yang terdapat pada sistem deteksi helm dapat berjalan sesuai dengan perancangan yang telah dibuat. Pengujian melalui behavioral ini dilakukan dengan cara memasukkan data pada software dengan dibandingkan pada *output* yang dihasilkan oleh software pada sistem deteksi helm.

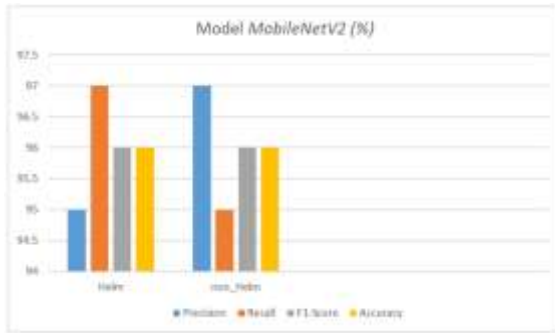
Tabel 1. Pengujian BlackBox pada uji Input Video Real-Time

Skenario Uji	Hasil Yang Diharapkan	Hasil Uji Percobaan	
		Y	T
Tidak memakai helm	Sistem menampilkan label pada objek yaitu non helm	Y	
Memakai helm	Sistem menampilkan label pada objek yaitu helm	Y	
Pengendara memakai hijab pada kepala	Sistem menampilkan label pada objek yaitu non helm	Y	
Pengendara memakai topi pada kepala	Sistem menampilkan label pada objek yaitu non helm	Y	
Pengendara memakai kopiah pada kepala	Sistem menampilkan label pada objek yaitu non helm	Y	
Pengendara hanya menggunakan masker	Sistem menampilkan label pada objek yaitu non helm	Y	
Pengendara menggunakan helm dan masker	Sistem menampilkan label pada objek yaitu helm		T

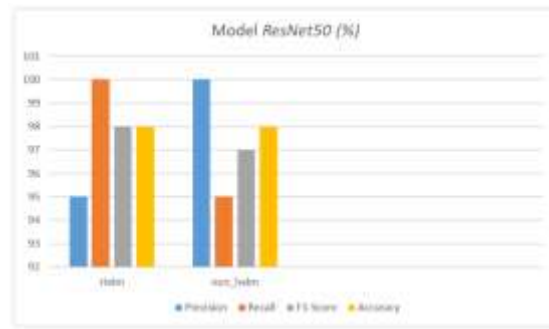
2. Pengujian Model Arsitektur Convolutional Neural Network

Sistem deteksi helm ini menggunakan *dataset* yang sudah di seleksi sesuai klasifikasi masing-masing dan memproses *dataset* menggunakan algoritma *Convolutional Neural Network* sehingga menjadikan *dataset* tersebut menjadi model untuk membantu sistem *machine learning* dalam mengeksekusi klasifikasi secara *real-time*. Untuk implementasi dalam membuat sebuah model, penulis menggunakan bahas pemograman *python* dan *library Deep learning Keras* penulis melakukan sebuah komparasi antara beberapa arsitektur untuk mengetahui arsitektur CNN

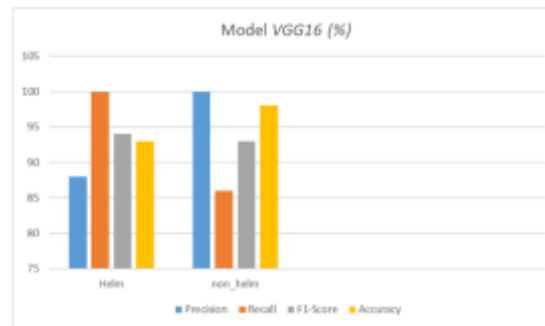
manakah yang lebih efektif dalam mengklasifikasi. Penulis menggunakan *MobileNetV2*, *ResNet50*, *VGG16* untuk melakukan komparasi. Dalam sistem pembuatan model klasifikasi memiliki batch size 32 dan penulis menggunakan epoch 20. Klasifikasi model pada Gambar 7 (a) di hasilkan dengan perintah pada kode program yang mempunyai nama *scikit-learn's classification_report*. Untuk membaca laporan klasifikasi pada Gambar 6 akan terbaca seperti berikut : kelas helm dan kelas non_helm memiliki presisi masing-masing 0,95 dan 0,97. Yaitu, dari hal-hal yang diklasifikasikan model sebagai kelas helm, hanya 95% yang benar-benar ada; dari hal-hal yang diklasifikasikan model sebagai kelas non_helm, 97% yang benar-benar ada; Presisi pada *macro average* adalah 0,96, dan *weighted average* adalah 0,96. Model memiliki *macro average* dan *weighted average* yang sama. Model ini menggunakan arsitektur *MobileNetV2*.



(a) Grafik Model MobileNetV2



(b) Grafik Model ResNet50



(c) Grafik Model VGG16
 Gambar 7. Grafik Hasil Uji

Klasifikasi model pada Gambar 7 (b) dihasilkan dengan perintah pada kode program yang mempunyai nama *scikit-learn's classification_report*. Untuk membaca laporan klasifikasi pada Gambar 7 akan terbaca seperti berikut : kelas helm dan kelas non_helm memiliki presisi masing-masing 0,95 dan 1,00. Yaitu, dari hal-hal yang diklasifikasikan model sebagai kelas helm, hanya 95% yang benar-benar ada; dari hal-hal yang diklasifikasikan model sebagai kelas non_helm, 100% yang benar-benar ada; Presisi pada *macro average* adalah 0,98, dan *weighted average* adalah 0,98. Model memiliki *macro average* dan *weighted average* yang sama. Model ini menggunakan arsitektur *ResNet50*.

Klasifikasi model pada Gambar 7 (c) dihasilkan dengan perintah pada kode program yang mempunyai nama *scikit-learn's classification_report*. Untuk membaca laporan klasifikasi pada Gambar 8 akan terbaca seperti berikut : kelas helm dan kelas non_helm memiliki presisi masing-masing 0,88 dan 1,00. Yaitu, dari hal-hal yang diklasifikasikan model sebagai kelas helm, hanya 88% yang benar-benar ada; dari hal-hal yang diklasifikasikan model sebagai kelas non_helm, 100% yang benar-benar ada; Presisi pada *macro average* adalah 0,94, dan *weighted average* adalah 0,94. Model ini menggunakan arsitektur *VGG16*. Untuk *epoch* yang di gunakan adalah 20 dan untuk *batch size* sebesar 32. Untuk iterasi dari model di dapati dengan hasil sebesar 25 dari persamaan (1).

$$\text{Iterasi} = \frac{\text{Total banyaknya dataset}}{\text{batch size}} \quad (1)$$

$$\text{Iterasi} = \frac{818}{32} = 25$$

Dengan demikian, model yang sudah terproses memiliki 25 iterasi dalam 1 *epoch*, jika di hitung kembali dengan *epoch* yang sudah tersedia yaitu 20 maka secara keseluruhan model akan memiliki 25 X 20 = 500 iterasi. Menurut beberapa artikel, untuk pola pada *plot* model *MobileNetV2* memiliki model yang cenderung bagus atau bisa di sebut Good Fit. Untuk *plot* model *VGG16* memiliki model yang *overfitting*. *Overfitting* terjadi saat kita melatih model *machine learning* yang terlalu pas dengan set pelatihan. Akibatnya, model belajar terlalu baik pada data pelatihan, tetapi gagal membuat prediksi yang baik pada data yang tidak terlihat. Untuk titik data yang tidak terlihat dalam pelatihan, model yang terlalu pas dapat menghasilkan hasil akurasi yang rendah, sehingga mengarah pada keputusan yang kurang optimal. Untuk *plot* model *ResNet50* memiliki model yang *underfitting*. *Underfitting* terjadi saat model *machine learning* tidak disetel dengan baik ke set pelatihan. Model yang dihasilkan tidak menangkap hubungan antara *input* dan *output* dengan cukup baik. Oleh karena itu, tidak menghasilkan prediksi yang akurat, bahkan untuk *dataset* pelatihan. Akibatnya, model *underfitted* menghasilkan hasil buruk yang mengarah pada keputusan kesalahan tinggi, seperti model *overfitted*. Pada Gambar 8, dengan menggunakan *matplotlib* pada hasil model yang sudah di proses memperlihatkan *plot* dari model tersebut, *plot* tersebut menampilkan *train_loss*, *val_loss*, *train_acc*, *val_acc*, dan pada masing-masing gambar memiliki pola garis yang berbeda-beda. Maka sebab itu, untuk model yang cocok untuk sistem deteksi ini adalah *MobileNetV2* yang memiliki model yang cenderung bagus atau *good fit*.



(a) Grafik Model MobileNetV2



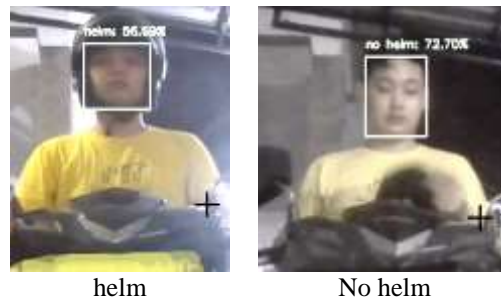
(b) Grafik Model ResNet50



(c) Grafik Model VGG16

Gambar 8. Training And Loss Menggunakan MobileNetV2, ResNet50, dan VGG16

Berikut ini adalah hasil implementasi dari program sistem deteksi helm menggunakan model yang sudah di proses seperti pada Gambar 9.



Gambar 9. Sample hasil identifikasi citra

Dari 116 percobaan deteksi, sistem dengan menggunakan model arsitektur *MobileNetV2* didapati mendeteksi orang menggunakan helm dengan hasil benar sebesar 93 dan hasil salah sebesar 23 sehingga untuk mengetahui nilai akurasi dari hasil deteksi helm yang benar secara klasifikasi, maka kita dapat hasil akurasi sebesar 80% dari 116 data tes yang di ujikan. Akurasi di dapat dengan hasil percobaan dengan memakai wajah dari penulis, akan tetapi sistem deteksi belum mampu untuk mendeteksi jika subjek membelakangi kamera dan hanya mampu mendeteksi jika subjek menghadap ke kamera.

$$\text{Akurasi} = \frac{\text{Hasil Benar}}{\text{Jumlah Data Testing}} \times 100\%$$

$$\text{Akurasi} = \frac{93}{116} \times 100\%$$

$$\text{Akurasi} = 80\%$$

5. KESIMPULAN

Sistem deteksi helm yang di kembangkan pada tugas ini dapat mendeteksi pada area kepala yang memberinya label helm dan non helm. Sistem ini mempergunakan model gambar yang dapat mencapai akurasi 80%. Ini karena model menggunakan 2 kelas dalam fase pelatihan, dengan masing-masing setiap kelas berjumlah 418 gambar untuk helm dan 400 gambar untuk tidak menggunakan helm. Pendeteksian pada wajah tidak dapat di lakukan jika objek bergerak terlalu cepat, dan posisi kamera juga mempengaruhi serta pencahayaan juga mempengaruhi. Berikut adalah kesimpulan bisa di tarik:

- Penelitian membangun sistem deteksi menggunakan algoritma Convolutional Neural Network dan pengimplementasian untuk video stream menggunakan MobileNetV2.
- Sistem deteksi dapat mengklasifikasi dan mendeteksi lebih dari satu orang menggunakan helm atau tidak menggunakan helm dan orang yang menggunakan tutup kepala selain helm.
- Sistem deteksi ini memiliki tingkat akurasi 80% untuk mendeteksi keadaan bagian kepala seseorang.
- Sistem ini masih dapat di kembangkan kembali menjadi lebih baik jika dataset yang digunakan adalah dataset yang berkualitas dan relevan.
- Penulis melakukan komparasi terhadap 3 arsitektur Convolutional Neural Network yaitu MobileNetV2, ResNet50, dan VGG16.

6. SARAN

Saran untuk penelitian selanjutnya adalah memperbanyak jumlah citra dengan latar belakang berbeda yang memungkinkannya meningkatkan akurasi di dalam kelas, dapat menambahkan GUI dan fitur lainnya supaya dapat di implementasikan secara luas, serta mempergunakan alat uji seperti webcam atau kamera yang memadai supaya mendapat hasil maksimal.

DAFTAR PUSTAKA

- [1] K. Shan, J. Guo, W. You, D. Lu, and R. Bie, "Automatic facial expression recognition based on a deep convolutional-neural-network structure," *Proc. - 2017 15th IEEE/ACIS Int. Conf. Softw. Eng. Res. Manag. Appl. SERA 2017*, pp. 123–128, 2017.
- [2] H. Lin, J. D. Deng, D. Albers, and F. W. Siebert, "Helmet Use Detection of Tracked Motorcycles Using CNN-Based Multi-Task Learning," *IEEE Access*, vol. 8, no. 3, pp. 162073–162084, 2020.
- [3] S. Rakhmadani, "Analisis Penerapan E-Tilang Dalam Mewujudkan Good Governance Di Indonesia," *Pros. SNA PP2017 Sos. Ekon. dan Hum.*, vol. 7, no. 3, pp. 663–671, 2017.
- [4] S. Chen, W. Tang, T. Ji, H. Zhu, Y. Ouyang, and W. Wang, "Detection of Safety Helmet Wearing Based on

- Improved Faster R-CNN,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–7.
- [5] L. Shine and J. C. V., “Automated detection of helmet on motorcyclists from traffic surveillance videos: a comparative analysis using hand-crafted features and CNN,” *Multimed. Tools Appl.*, vol. 79, no. 19–20, pp. 14179–14199, May 2020.
- [6] P. Samant and R. Agarwal, “Machine learning techniques for medical diagnosis of diabetes using iris images,” *Comput. Methods Programs Biomed.*, vol. 157, pp. 121–128, Apr. 2018.
- [7] 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,” *Adv. Civ. Eng.*, vol. 2020, 2020.
- [8] C. Vishnu, D. Singh, C. K. Mohan, and S. Babu, “Detection of motorcyclists without helmet in videos using convolutional neural network,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3036–3041.
- [9] C. Raras and A. Widiawati, “Automatic RoI dan Active Contour untuk Deteksi Penggunaan Helm pada Pengendara Sepeda Motor,” *JISA (Jurnal Inform. dan Sains)*, vol. 02, no. 02, pp. 41–46, 2019.
- [10] K. Gunadi, E. Setyati, and J. S. Surabaya, “Deteksi Helm pada Pengguna Sepeda Motor dengan Metode Convolutional Neural Network,” *Jurnal INFRA*, vol. 8, no. 1, 2020.
- [11] B. Widodo, H. A. Armanto, and E. Setyati, “Deteksi Pemakaian Helm Proyek Dengan Metode Convolutional Neural Network,” *J. Intell. Syst. Comput.*, vol. 3, no. 1, pp. 23–29, Apr. 2021.
- [12] Y. Kulkarni, S. Bodkhe, A. Kamthe, and A. Patil, “Automatic number plate recognition for motorcyclists riding without helmet,” in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 2018, pp. 1–6.
- [13] M. M. Thaha, K. P. M. Kumar, B. S. Murugan, S. Dhanasekeran, P. Vijayakarthish, and A. S. Selvi, “Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images,” *J. Med. Syst.*, vol. 43, no. 9, 2019.
- [14] D. L. B, H. Ma, Z. Xiong, and F. Wu, *MultiMedia Modeling*, vol. 10705. Cham: Springer International Publishing, 2018.
- [15] A. H. Mawaddah, C. Atika Sari, D. R. Ignatius Moses Setiadi, and E. Hari Rachmawanto, “Handwriting Recognition of Hiragana Characters using Convolutional Neural Network,” in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2020, pp. 79–82.
- [16] D. Yashunin, T. Baydasov, and R. Vlasov, “MaskFace: multi-task face and landmark detector,” May 2020.
- [17] W. Wu, Y. Yin, X. Wang, and D. Xu, “Face detection with different scales based on faster R-CNN,” *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 4017–4028, 2019.
- [18] A. Chavda, J. Dsouza, S. Badgular, and A. Damani, “Multi-Stage CNN Architecture for Face Mask Detection,” *2021 6th Int. Conf. Converg. Technol. I2CT 2021*, pp. 1–8, 2021.