Jurnal Teknik Informatika (JUTIF)
Vol. 3, No. 2, April 2022, hlm. 421-427

DOI: https://doi.org/10.20884/1.jutif.2022.3.2.219
p-ISSN: 2723-3863
e-ISSN: 2723-3871

RECOGNITION OF HUMAN FACES IN VIDEO CONFERENCE APPLICATIONS USING THE CNN PIPELINE

Evan Tanuwijaya[*1], Reinaldo Lewis Lordianto[2], Reiner Anggriawan Jasin[3]

[1,2,3]Informatika, Fakultas Teknologi Informasi, Universitas Ciputra Surabaya, Indonesia
Email: [1]evan.tanuwijaya@ciputra.ac.id, [2]rlordianto@student.ciputra.ac.id , [3]ranggriawan@student.ciputra.ac.id

(Naskah masuk: 16 Maret 2022, Revisi: 17 Maret 2022, Diterbitkan: 25 April 2022)

Abstract

The COVID-19 pandemic has forced daily face-to-face activities to be carried out online using video conferencing applications. To record participant participation in meetings using a video conference application, an online form application is used. However, participants sometimes do not see this and are often missed due to the large number of incoming chats. Therefore, the use of face detection for attendance using a combination of CNN to detect all the faces in a video conference using YOLO Face and CNN to recognize the owner of a face using Smaller VGG in a pipeline will make it easier to recognize participants who are present at the video conference. The results of the Smaller VGG training are obtained, namely the loss value of 0.059, the accuracy value is 0.995, the recall value is 0.994, the precision value is 0.996. Meanwhile, for the validation phase of the model, the loss value is 0.497, the accuracy value is 0.979, the recall value is 0.979 and the precision value is 0.981. In terms of training duration, the smaller VGG has a duration of 4 minutes and 16 seconds. The Smaller VGG model was combined with YOLO to create a CNN pipeline and was successful in recognizing the faces of video conference participants

Keywords: *Convolution Neural Network, Deep Learning , Face Recognition, YOLO*

PENGENALAN WAJAH MANUSIA PADA APLIKASI VIDEO CONFERENCE MENGGUNAKAN METODE PIPELINE CNN

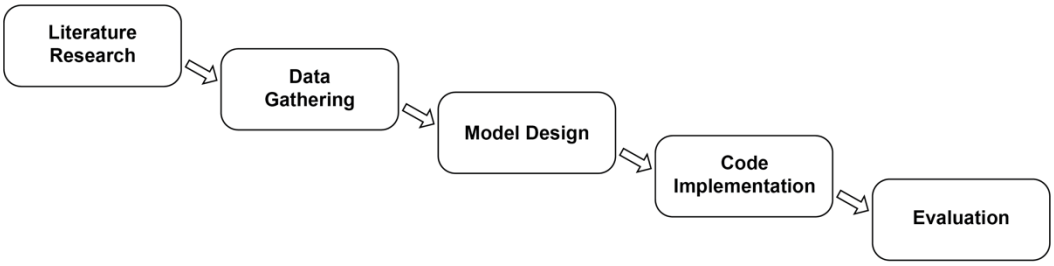
Abstrak

Pandemi Covid 19 membuat kegiatan yang sehari-hari secara tatap muka harus dilakukan online menggunakan aplikasi video conference. Mencatat partisipasi peserta dalam pertemuan menggunakan aplikasi video conference, digunakan aplikasi form online. Akan tetapi, peserta terkadang tidak melihat hal tersebut dan bahkan sering terlewati akibat banyaknya chat yang masuk. Oleh sebab itu, pemanfaatan deteksi wajah untuk absensi kehadiran menggunakan gabungan CNN untuk deteksi semua wajah yang ada pada video conference menggunakan YOLO Face dan CNN untuk mengenali pemilik wajah menggunakan Smaller VGG secara pipeline akan mempermudah dalam mengenali peserta yang hadir pada video conference. Hasil training Smaller VGG didapatkan yaitu nilai loss sebesar 0,059 nilai akurasi yaitu 0,995 nilai recall 0,994 nilai precision 0,996. Sementara untuk fase validasi dari model didapatkan nilai loss 0,497 nilai akurasi 0,979 nilai recall adalah 0,979 dan nilai precision 0,981. Dari sisi durasi training smaller VGG memiliki durasi 4 menit dan 16 detik. Model Smaller VGG dikombinasikan dengan YOLO untuk membuat CNN pipeline dan berhasil dalam mengenali wajah peserta video conference

Kata kunci: *Convolution Neural Network, Deep Learning , Pengenalan Wajah, YOLO*

1. PENDAHULUAN

Penggunaan komputer membantu mendeteksi objek sudah sangat sering digunakan oleh berbagai macam bidang salah satu pemanfaatannya adalah deteksi wajah. Deteksi wajah sudah banyak diterapkan di berbagai macam aplikasi seperti kamera di jalan raya, menjaga keamanan rumah, absensi di dalam kelas dan masih banyak lagi [1]. Di saat pandemi Covid 19 ini, kegiatan dilakukan secara daring menggunakan aplikasi video *conference* seperti Zoom atau Google Meet. Untuk mencatat kehadiran dari peserta, penyelenggara kegiatan biasa menggunakan aplikasi form seperti Google Form yang disebarkan menggunakan *chat*. Akan tetapi, peserta terkadang tidak melihat hal tersebut dan bahkan sering terlewati akibat banyaknya *chat* yang masuk ditambah lagi untuk mengisi form, peserta perlu menginputkan data-data dengan cara mengetikan data seperti email maupul



Gambar 1. SDLC Waterfall model

hal-hal lain yang dirasa kurang tepat untuk melakukan absensi.. Salah satu pemanfaatan yang dapat diterapkan adalah deteksi wajah untuk absensi kehadiran [2], [3] karena dengan menggunakan deteksi wajah, data nama yang dibutuhkan bisa langsung didapatkan

Sudah banyak penelitian dilakukan untuk mendeteksi wajah menggunakan komputer. Metode yang digunakan meliputi, 2D-image *principal component analysis*, *Local binary patterns*, *Histogram of oriented gradients* HOG, Neural Network dan masih banyak lagi [4]. *Neural network* merupakan salah satu metode yang popular digunakan oleh banyak peneliti karena dapat mengolah jumlah dimensi atribut yang banyak pada sebuah data. Dengan menerapkan *deep learning* pada *neural network*, *Convolution Neural Network* mampu melakukan deteksi objek atau wajah dengan memproses citra secara langsung [5]. *Convolution neural network* mampu melakukan ekstraksi fitur dengan mengolah citra dengan berbagai macam *kernel* yang di tumpuk pada bagian *convolution* layernya, kemudian melakukan klasifikasi dengan neural network untuk menghasilkan prediksi yang sesuai [6].

Convolution neural network merupakan salah satu metode *machine learning* dengan menerapkan konsep *deep learning* yang dapat digunakan untuk melakukan klasifikasi suatu objek pada citra [7], [8]. *Convolution neural network* atau CNN dapat melakukan klasifikasi objek pada citra menggunakan *kernel* atau *convolution* layer untuk melakukan ekstraksi fitur kemudian di proses menggunakan *neural network* untuk melakukan klasifikasi. *Convolution* layer sendiri memiliki beberapa komponen. Dua komponen yang umum digunakan pada *convolution* layer adalah *kernel* untuk melakukan ekstraksi fitur dengan cara nilai *pixel* pada citra dihitung dengan nilai yang ada pada *kernel* kemudian diterapkan fungsi aktivasi untuk menentukan keaktifan dari nilai tersebut [9]. Komponen umum yang kedua adalah *pooling* layer dimana layer ini akan mengecilkan dimensi citra hasil pemrosesan *kernel* dengan cara mengambil nilai tertinggi atau rata-rata dari $n \; x \; n$ pixel. YOLO-face merupakan CNN yang dapat melakukan deteksi objek dimana model CNN ini sudah dilakukan *pre-trained* untuk dapat mendeteksi wajah pada citra. YOLO-face sendiri merupakan pengembangan dari *You Only Look Once* [10] yang dilatih menggunakan data WIDER FACE *datasets* untuk dapat mendeteksi wajah pada citra [11]. YOLO merupakan *state-of-the-art* dalam objek deteksi yang dapat melakukan deteksi secara *real-time* [12].

Berdasar permasalahan di atas, peneliti ingin menggabungkan CNN untuk deteksi semua wajah yang ada pada video *conference* dan CNN untuk mengenali pemilik wajah secara *pipeline*. Dengan menggabungkan kedua CNN tersebut dengan metode *pipeline*, akan mempermudah peneliti dalam melakukan proses *training*. Dimana CNN untuk mendeteksi semua wajah pada video *conference* dapat menggunakan YOLO-*Face* [11], kemudian hasil dari YOLO-*face* akan di proses menggunakan CNN kedua yang lebih sederhana untuk melakukan identifikasi wajah seseorang.

2. METODE PENELITIAN

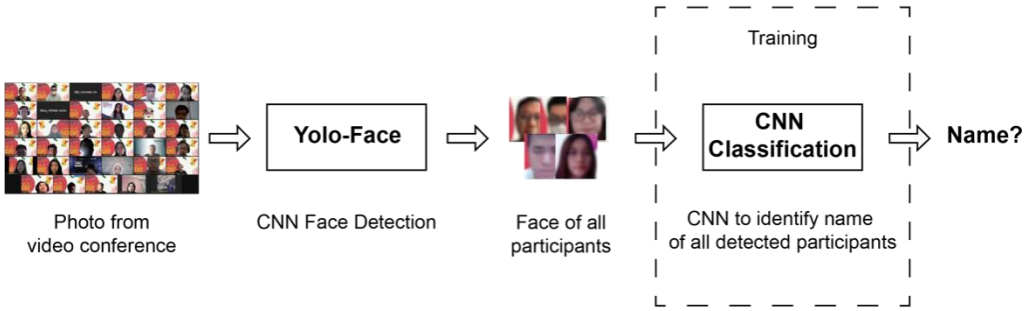
Dalam pengembangan sebuah perangkat lunak, *System Development Life Cycle* merupakan metodologi yang umum digunakan. SDLC memiliki fase-fase seperti perencanaan, analisis, desain, implementasi dan evaluasi sebagai kerangkanya [13]. Pada penelitian ini, SDLC model *waterfall* yang akan digunakan dengan detail yang dapat dilihat pada Gambar 1.

Tahap awal dari penelitian ini adalah melakukan *literature research* dimana, peneliti akan mempelajari *convolution neural network* yaitu *You Only Look Once* (YOLO-*face*) untuk melakukan deteksi semua wajah pada foto video *conference* kemudian mencari informasi terkait model *convolution neural network* untuk melakukan objek klasifikasi.

Tahap kedua adalah data *gathering*. Data *gathering* ini akan mengumpulkan foto-foto wajah yang akan dideteksi. Data yang akan dikumpulkan adalah data wajah dari ketiga peneliti dan 10 orang lainnya. Masing-masing orang-orang akan dikumpulkan minimal 100 wajah dengan menggunakan website *Teachable Machine* sehingga dihasilkan *datasets* wajah yang siyap untuk di proses untuk fase *training convolution neural network.*

Pada tahap model desain, akan dibangun model berupa *pipeline*. Model ini akan menggabungkan

dua buat *convolution neural network* dengan konsep kemudian di modifikasi untuk mengklasifikasikan



Gambar 2. Model Design Pipeline CNN

seperti Gambar 2. Foto video *conference* dimasukkan ke dalam YOLO-*face* untuk dideteksi seluruh wajah yang ada pada video *conference* sehingga didapatkan potongan bersih wajah-wajah yang ada pada video *conference*. YOLO-*face* merupakan *pre-trained* CNN yang hanya mendeteksi wajah saja tetapi tidak dapat mengenali identitas dari wajah tersebut. Untuk melengkapi hal tersebut, CNN klasifikasi akan mengidentifikasi potongan wajah yang terdeteksi dari YOLO-*face* untuk diketahui nama dari masing-masing potongan wajah tersebut. Peneliti akan membentuk tiga buah CNN klasifikasi dari hasil *literature research* untuk dibandingkan pada fase evaluasi untuk mencari model terbaik.

Tahap keempat adalah *code implementation* dimana tahap ini akan membangun kode program untuk masing-masing komponen. Untuk YOLO-face, peneliti menggunakan pre-*trained* CNN sehingga bisa langsung di implementasi. Sementara CNN untuk klasifikasi, terdapat fase *training* dimana CNN harus diubah parameternya berdasarkan hasil *training* agar dapat mengklasifikasikan foto hasil YOLO-*face*. Kontribusi dari penelitian ini adalah model *pipeline* dan model CNN klasifikasi yang didapatkan dari *literature research*. Keunggulan dari metode *pipeline* ini adalah peneliti tidak perlu melakukan *training* untuk semua model CNN karena menggunakan pre-*trained* dari YOLO-*face* dan tidak perlu melakukan *marking datasets* hanya perlu mengelompokkan data foto sesuai dengan kelas atau nama dari pemilik wajah tersebut.

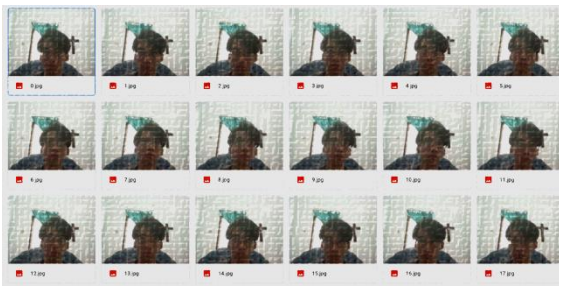
Tahap terakhir dari penelitian ini adalah evaluasi dimana ketiga model tersebut akan dibandingkan akurasi, *precision*, dan *recall* untuk menentukan model yang terbaik pada data foto video *conference*.

3. HASIL DAN PEMBAHASAN

Penelitian ini dimulai dengan *literature research* mendapatkan beberapa model untuk mengklasifikasikan wajah manusia. Pada penelitian pertama berasal dari [14] dimana pada penelitian ini terdapat sebuah model *convolution neural network* yang dibuat berdasarkan model VGG 16 yang rempah-rempah Indonesia. Penelitian kedua didapatkan model arsitektur *Venturi* dimana model tersebut berfokus pada *hidden* layer yang mengecil pada bagian *convolution* layer dan Kembali membesar saat masuk ke dalam *fully connected* layer [15] yang digunakan untuk mengklasifikasikan ekspresi wajah pada manusia. Penelitian ketiga yang modelnya digunakan sebagai pembanding adalah [16] menggunakan low level *feature* yang digabungkan dengan *high* level filter yang untuk membentuk sebuah **deep** *learning* yang dipakai untuk mengklasifikasikan kapal laut. Dari ketiga **literature** tersebut, model yang didapatkan akan dibandingkan akurasi, *presicion*, *recall*, dan *loss* untuk menentukan model yang tepat sebagai model klasifikasi dari YOLO.

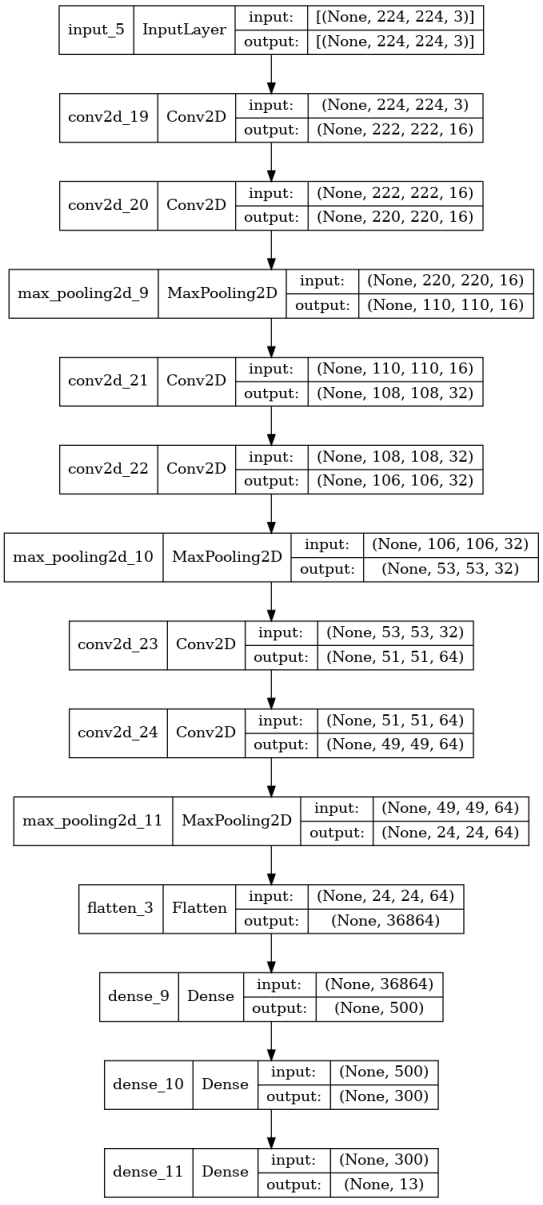
Pada tahap data *gathering*, peneliti mengumpulkan 13 orang untuk di ambil data wajahnya. Data diambil menggunakan website *Teachable Machine*. Data yang didapatkan berukuran 224 x 224 ber-*channel* RGB sebanyak 200 data masing-masing orang seperti gambar 3. Data wajah kemudian dikelompokkan berdasarkan nama folder.

Model yang digunakan dalam penelitian ini adalah ketiga model yang didapatkan dari tahap *literature* yang kemudian akan dibandingkan hasil *training* dan testing dengan *dataset*. Model pertama adalah *Smaller* VGG yang dapat dilihat pada gambar 4. Model ini menggunakan *input* dengan ukuran 224 x 224 x 3 dilanjutkan dengan 2 buah *convolution* layer dengan ukuran 3 x 3 sebanyak 16 dengan *activation* layer *relu*. Kemudian terdapat *maxpooling* layer dengan *pool size* sebesar 2 dan *stride* 2. *Convolution* layer selanjutnya terdapat 2 buah dengan ukuran 3 x 3 sejumlah 32 dengan *activation* layer *relu* dan *maxpooling* dengan *pool size* sebesar 2 dan *stride* 2. Layer *convolution* selanjutnya menggunakan ukuran 3 x 3 dengan jumlah 64 *kernel*



Gambar 3. Dataset Wajah

dan *activation relu* sebanyak 2 layer dan *maxpooling* dengan *pool size* sebesar 2 dan *stride* 2. Kemudian dimasukkan ke dalam *flatten* layer untuk di proses pada *fully connected* layer yang memiliki 2 *hidden* layer yang masing-masing *hidden* layer memiliki 500 dan 300 neuron. Untuk *output* layer menggunakan *activation softmax* berjumlah 13 kelas.

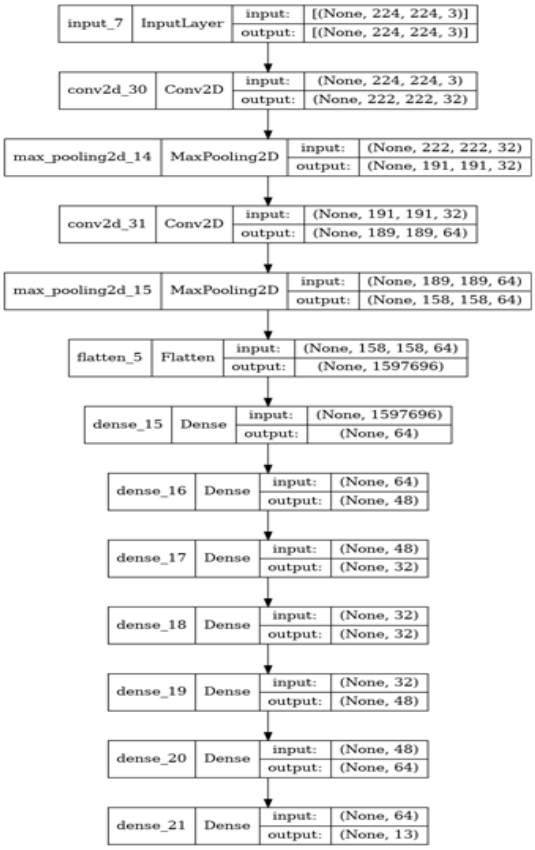


Gambar 4. Model Smaller VGG

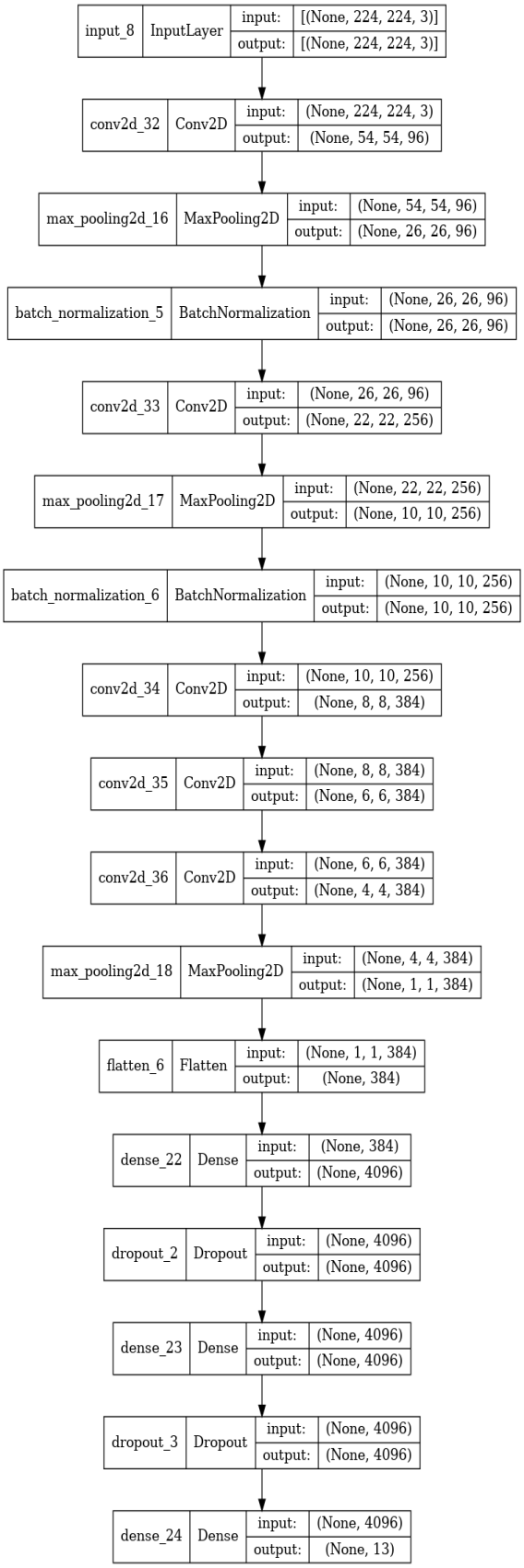
Model kedua adalah model *venturi* dimana model ini berfokus pada *fully connected layer* yang lebih banyak dibandingkan jumlah *convolution* nya pada gambar 5. Pada model *venturi* ini menerima *input* dengan ukuran 224 x 224 x 3 kemudian terdapat *convolution* layer dengan ukuran 3 x 3 sebanyak 32 *kernel* dengan *activation function relu* setelah itu terdapat *maxpooling* layer dengan ukuran

1 x 1 *stride* 1. Layer kedua terdapat *convolution* layer sebesar 3 x 3 dengan jumlah layer 64 dan *activation function relu* dan *maxpooling* dengan *stride* 1. Kemudian dilakukan *flatten* dan kemudian dimasukkan ke dalam fully *connected* layer dengan 6 *hidden* layer yang memiliki neuron sebanyak 64, 48, 32, 32, 48, dan 64 kemudian diakhiri dengan *output* layer dengan jumlah 13 neuron dan *activation function softmax*.

Model ketiga diambil dari model klasifikasi kapal dimana menggabungkan antara *low fitur* dengan *high fitur* pada gambar 6. Model ini memiliki *input* dengan ukuran 224 x 224 x 3. *Convolution* layer pertama memiliki ukuran 11 x 11 dengan *stride* 4 dengan jumlah *kernel* sebanyak 96 dan *activation function relu*. Layer *maxpooling* dengan ukuran 3 x 3 dengan *stride* 2 dilanjutkan dengan *batch normalization*. *Convolution* layer kedua memiliki ukuran 5 x 5 dengan jumlah *kernel* 256 dan *stride* 1 dengan *activation function relu* dilanjutkan dengan *maxpooling* ukuran 3 x 3 dan *stride* 2 selanjutnya diberikan layer *batch normalization*. Layer ketiga, empat dan lima memiliki parameter yang sama yaitu jumlah *kernel* 384 dengan ukuran 3 x3 dengan *stride* 1 dan *activation function relu*. Kemudian di *flatten* untuk di proses ke dalam *fully connected layer* dengan 2 *hidden* layer dengan jumlah parameter 4096 dan droput sebesar 0,5. *Output* layer memiliki jumlah neuron sebanyak 13 dengan *activation function softmax*.

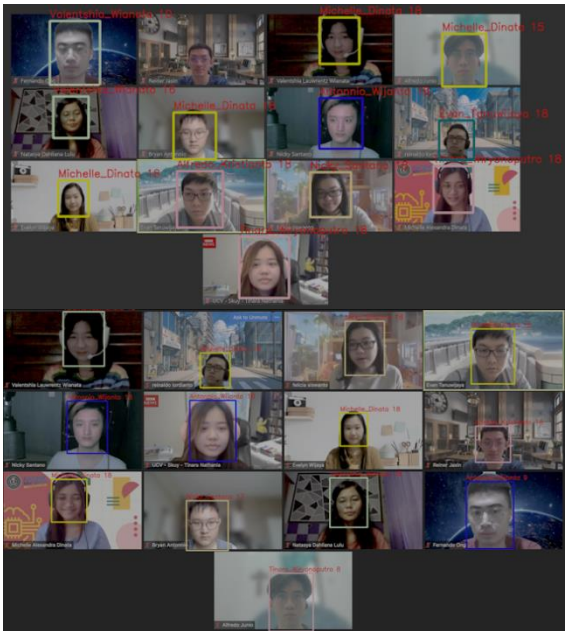


Gambar 5. Model *Venturi*



Gambar 6. Model CNN

Dari ketiga model tersebut dilatih dengan komputer yang memiliki *Graphic Processor Unit*



Gambar 7. Hasil Deteksi YOLO dan Smaller VGG

Quadro RTX 4000 yang *memilki* GRAM 8 GB. Model dilatih dengan *hyperparameter optimizer* adam, *loss function categorical crossentropy*, dan *epoch* sebanyak 100 dan *batch size* sebesar 40. Didapatkan hasil *training* pada tabel 1 dimana evaluasi model yang dipakai sebagai pembanding adalah durasi latih, akurasi, *loss*, *precision*, *recall*, *validation loss*, *validation* akurasi, *validation recall*, dan *validation precision*. Didapatkan durasi tercepat saat *training* adalah model ketiga yaitu model dari CNN deteksi kapal dengan nilai 3 menit dan 34 detik. Akan tetapi dari akurasi, *precision*, *recall*, dan nilai *loss* yang terbaik dari sisi training dan validasi *training* dimiliki oleh model pertama yaitu *smaller* VGG dengan nilai *loss* 0,059 dan validasi loss 0,497 Akurasi sebesar 0,995 dan validasi akurasi sebesar 0.979. Nilai *precision* memiliki nilai 0.996 dan validasi *precision* sebesar 0.981 sementara *recall* miliki nilai 0.994 dan validasi *recall* yaitu 0,979. Untuk model kedua memiliki total durasi yang *relative* lama karena memiliki jumlah *hidden* layer yang cukup banyak sehingga proses generalisasi semakin lama sementara *fitur* yang didapatkan dari *convolution* layer memiliki *fitur* yang kurang sehingga saat masuk ke dalam *fully connected* layer tidak dapat mendeskripsikan dengan baik.

Dari hasil perbandingan ketiga model tersebut, model *smaller* VGG dipakai sebagai model CNN untuk klasifikasi wajah pada aplikasi video *conference*. Berdasarkan gambar 2, Citra dari hasil

Gambar 8. Hasil Bounding Box dari YOLO

Tabel 1. Perbandingan Model

	Smaller VGG	**Venturi Model**	**CNN Klasifikasi Kapal**
Train Duration	04:16	33:51	03:34
Loss	0,059	2,825	0,681
Akurasi	0,995	0,226	0,905
Recall	0,994	0,153	0,889
Precision	0,996	0,977	0,928
Val Loss	0,497	2,074	6,395
Val Akurasi	0,979	0,242	0,882
Val Recall	0,979	0,162	0,869
Val Precision	0,981	0,976	0,901

screenshot video *conference* di proses terlebih dahulu oleh YOLO untuk didapatkan *bounding box*. Proses deteksi YOLO menggunakan *single neural network* untuk mendeteksi *bounding box* dan kelas dari *bounding box* tersebut. YOLO akan membagi citra menjadi beberapa grid *cell* yang masing-masing *grid cell* akan di prediksi koordinat dan kelas yang ada pada citra tersebut menggunakan *darknet* 53. Hasil dari *darknet* tersebut akan memiliki banyak sekali *bouding box* sehingga perlu dieliminasi *bounding box* yang kurang sesuai menggunakan *non-maximun suppresion* dan *Intersection over Union* [11]. Hasil *bounding box* kemudian akan diklasifikasikan dengan *smaller* VGG untuk diketahui nama dari masing-masing orang yang ada pada video *conference* tersebut.

Untuk hasil dari CNN *pipeline* ini dapat dilihat dari Gambar 7 dimana hasil *bounding box* akan di gambar pada citra dan di berikan nama di bagian atas berdasarkan hasil prediksi. Model CNN *pipeline* ini berhasil dalam menandai wajah manusia yang ada pada citra dan berhasil memprediksi nama dari masing-masing *bounding box*. Akan tetapi terdapat beberapa kekeliruan dalam prediksi. Hal ini disebabkan oleh adanya perbedaan ukuran gambar dan keadaan gambar dari *dataset* dengan hasil *bounding box* pada YOLO. Pada Gambar 8 merupakan salah satu hasil *face detection* dari YOLO yang mendeteksi wajah pada gambar dan memberikan *bounding box* di sekitar gambar. Jika diperhatikan dari Gambar 1 dimana *input*-an dari model klasifikasi memiliki *background* yang cukup lebar dibandingkan dengan gambar 8 yang berfokus pada wajah. Hal ini menyebabkan beberapa klasifikasi mengalami permasalahan dan menyebabkan kesalahan dalam prediksi.

4. KESIMPULAN

Penelitian pengenalan wajah pada aplikasi video *conference* berguna untuk mengidentifikasi peserta yang hadir menggunakan *convolution neural network*. CNN yang digunakan pada penelitian ini ada dua yang pertama CNN untuk mendeteksi wajah pada citra dan CNN kedua adalah CNN untuk mengklasifikasikan wajah yang dideteksi dari CNN pertama itu dengan nama dari masing-masing wajah. CNN pertama menggunakan YOLO *Face* yang sudah di latih dan menghasilkan sebuah *bounding box* yang menandai semua wajah yang ada dalam citra. CNN ke dua, peneliti membandingkan tiga buah model yaitu model *smaller* VGG, *Venturi* model, dan CNN untuk mengklasifikasikan kapal. Didapatkan klasifikasi terbaik terdapat pada model *smaller* VGG yaitu nilai *loss* sebesar 0,059 nilai akurasi yaitu 0,995 nilai *recall* 0,994 nilai *precision* 0,996. Sementara untuk fase validasi dari model didapatkan nilai *loss* 0,497 nilai akurasi 0,979 nilai *recall* adalah 0,979 dan nilai *precision* 0,981. Dari sisi durasi *training smaller* VGG memiliki durasi 4 menit dan 16 detik. Model *smaller* VGG di pakai untuk CNN klasifikasi dan berhasil mengklasifikasikan wajah hasil dari CNN pertama yaitu YOLO. Akan tetapi terdapat beberapa kekurangan dimana hasil klasifikasi tidak dapat memprediksi secara tepat hasil dari YOLO dikarenakan perbedaan data *training* dengan hasil *bounding box* dari YOLO yang memiliki *background* yang sedikit. Oleh sebab itu penelitian selanjutnya bisa menggunakan YOLO untuk membersihkan *dataset* yang dikumpulkan dari *Teachable Machine* dengan YOLO kemudian baru dilakukan proses *training* CNN klasifikasi.

DAFTAR PUSTAKA

[1] K. B. Pranav and J. Manikandan, "Design and Evaluation of a Real-Time Face Recognition System using Convolutional Neural Networks," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1651–1659, 2020, doi: 10.1016/j.procs.2020.04.177.

[2] B. Tej Chinimilli, A. Anjali, A. Kotturi, V. Reddy Kaipu, and J. Varma Mandapati, "Face Recognition based Attendance System using Haar Cascade and Local Binary Pattern Histogram Algorithm," *Proc. 4th Int. Conf. Trends Electron. Informatics, ICOEI 2020*, no. Icoei, pp. 701–704, 2020, doi: 10.1109/ICOEI48184.2020.9143046.

[3] H. Yang and X. Han, "Face recognition attendance system based on real-time video processing," *IEEE Access*, vol. 8, pp. 159143–159150, 2020, doi:

10.1109/ACCESS.2020.3007205.

[4] U. Jayaraman, P. Gupta, S. Gupta, G. Arora, and K. Tiwari, "Recent development in face recognition," *Neurocomputing*, vol. 408, pp. 231–245, 2020, doi: 10.1016/j.neucom.2019.08.110.

[5] B. Li and D. Lima, "Facial expression recognition via ResNet-50," *Int. J. Cogn. Comput. Eng.*, vol. 2, no. February, pp. 57–64, 2021, doi: 10.1016/j.ijcce.2021.02.002.

[6] Z. Chen and P. H. Ho, "Global-connected network with generalized ReLU activation," *Pattern Recognit.*, vol. 96, 2019, doi: 10.1016/j.patcog.2019.07.006.

[7] S. Khan, M. H. Javed, E. Ahmed, S. A. A. Shah, and S. U. Ali, "Facial recognition using convolutional neural networks and implementation on smart glasses," *2019 Int. Conf. Inf. Sci. Commun. Technol. ICISCT 2019*, pp. 1–6, 2019, doi: 10.1109/CISCT.2019.8777442.

[8] Y. Tian, "Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm," *IEEE Access*, vol. 8, pp. 125731–125744, 2020, doi: 10.1109/ACCESS.2020.3006097.

[9] J. Tang, Q. Su, B. Su, S. Fong, W. Cao, and X. Gong, "Parallel ensemble learning of convolutional neural networks and local binary patterns for face recognition," *Comput. Methods Programs Biomed.*, vol. 197, p. 105622, 2020, doi: 10.1016/j.cmpb.2020.105622.

[10] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, 2020.

[11] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, "YOLO-face: a real-time face detector," *Vis. Comput.*, 2020, doi: 10.1007/s00371-020-01831-7.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." Accessed: May 18, 2021. [Online]. Available: http://pjreddie.com/yolo/

[13] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, pp. 1–5, 2020.

[14] D. C. Khrisne and I. A. Suyadnya, "Indonesian Herbs and Spices Recognition using Smaller VGGNet-like Network," *2018 Int. Conf. Smart Green Technol. Electr. Inf. Syst.*, vol. 4, pp. 221–224, 2018.

[15] A. Verma, P. Singh, and J. S. Rani Alex, "Modified Convolutional Neural Network Architecture Analysis for Facial Emotion Recognition," *Int. Conf. Syst. Signals, Image Process.*, vol. 2019-June, pp. 169–173, 2019, doi: 10.1109/IWSSIP.2019.8787215.

[16] Q. Shi, W. Li, F. Zhang, W. Hu, X. Sun, and L. Gao, "Deep CNN with Multi-Scale Rotation Invariance Features for Ship Classification," *IEEE Access*, vol. 6, pp. 38656–38668, 2018, doi: 10.1109/ACCESS.2018.2853620.