

# Emotion Recognition using Convolutional Neural Network in Virtual Meeting Environment

Julando Omar<sup>1</sup>, Akmal Nusa Bhakti<sup>2</sup>, Axel Patria<sup>3</sup>

<sup>1, 2, 3</sup> Teknik Komputer, Universitas Multimedia Nusantara, Tangerang, Indonesia

[julando.omar@student.umn.ac.id](mailto:julando.omar@student.umn.ac.id)

[akmal.nusa@student.umn.ac.id](mailto:akmal.nusa@student.umn.ac.id)

[axel.patria@student.umn.ac.id](mailto:axel.patria@student.umn.ac.id)

Diterima dd mmmmm yyyy

Disetujui dd mmmmm yyyy

**Abstract** — Kajian ini dilakukan untuk memperkenalkan sebuah sistem *Emotion Recognition* yang menggunakan sebuah *Convolutional Neural Network* pada lingkungan *Virtual Meeting* untuk mendeteksi *feedback* nonverbal yang ada pada saat berkomunikasi. Kajian dimulai dengan melakukan proses *training* terhadap model CNN yang menggunakan library tensorflow-gpu versi 2.3.0 serta dataset FER-2013, dimana 80% data digunakan sebagai *training set* dan 20% digunakan sebagai *test set*. Model melakukan *training* sebanyak 430 *epochs* yang menghasilkan *accuracy* sebesar 73.86% dengan *loss* sebesar 1.42. Pada proses *classification*, algoritma *Haar-Cascade Classifier* digunakan untuk mendeteksi wajah pada gambar yang diinput menggunakan OpenCV. Kemudian model yang telah dibuat melakukan prediksi terhadap gambar yang telah di *pre-process*. Berdasarkan hasil kajian yang ada, dapat disimpulkan bahwa kajian telah memberikan hasil yang memuaskan dan diharapkan dapat membantu dalam mengetahui masukan nonverbal yang diberikan saat berkomunikasi dan juga dalam berbagai hal lainnya.

**Index Terms** - *Emotion Recognition, Convolutional Neural Network, Face Recognition, FER-2013, Virtual Meeting*

## I. PENDAHULUAN

Dalam kurun waktu setahun ini, dunia edukasi mengalami perubahan drastis dalam cara kita berkomunikasi. Akibat dari pandemi yang muncul, memaksa kita untuk merubah cara kita berkomunikasi dari tatap muka menjadi secara daring. Dari percakapan sederhana antara dua orang, proses belajar mengajar, hingga sebuah *meeting* penting yang dilakukan oleh sebuah perusahaan, semua terpaksa dilakukan secara daring. Menurut penelitian yang dilakukan oleh ISED, pada bulan April 2020, hampir 88% responden melakukan work from home, dan melakukan meeting dengan menggunakan aplikasi virtual meeting seperti *zoom meeting, google hangout* dan *webex*. [1]

Sistem komunikasi daring sendiri tidak terlepas dari beberapa kekurangan, salah satunya adalah *feedback* yang diberikan saat berkomunikasi.

*Feedback* verbal dan nonverbal yang biasanya dapat dilakukan saat berkomunikasi tatap muka mengalami hambatan dalam berkomunikasi daring. Ekspresi wajah terutama dapat menjadi komunikasi untuk mengutarakan perasaan kenyamanan, persetujuan maupun simpati kepada lawan bicara, tanpa harus adanya informasi tambahan dari anggota tubuh lainnya. [2]

Untuk mengatasi permasalahan ini, kami membuat sebuah *Emotion Recognition* yang menggunakan sebuah *Convolutional Neural Network* pada sebuah lingkungan *Virtual Meeting*, dengan tujuan untuk membantu memahami *feedback* nonverbal yang diberikan melalui ekspresi lawan bicara. Cara ini dilakukan dengan melakukan dua tahapan *training* dan *classification*.

Tahapan *training* merupakan tahapan dimana model CNN akan dipaparkan terhadap dataset FER-2013 yang berisikan berbagai ekspresi dari senang, sedih, marah, dst. Kemudian model akan dilakukan validasi pada tahapan *classification*, dimana model akan diuji terhadap input yang diberikan dan dihitung tingkat keakuratan dari model tersebut..

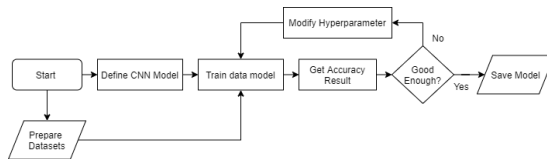
## II. KAJIAN LITERATUR

Studi mengenai implementasi dari *Convolutional Neural Network* untuk mengetahui ekspresi muka manusia sudah banyak dilakukan. Lasri, I. et al [3] mengusulkan metode menggunakan CNN untuk mengetahui ekspresi muka dari murid. Lasri, I. et al menggunakan algoritma *Haar Cascade* untuk melakukan ekstraksi wajah manusia dan menggunakan CNN yang ditraining menggunakan dataset FER-2013 untuk mengklasifikasi ekspresi wajah manusia. Metode tersebut berhasil untuk mencapai akurasi hingga 70%. Connie T. et al. [4] mengusulkan metode pengenalan ekspresi manusia dengan CNN yang digabungkan dengan Dense SIFT dan SIFT. metode ini berhasil untuk mengklasifikasikan ekspresi wajah manusia dengan akurasi sebesar 73.4%. D. Yang et. al. [5] mengajukan

metode pengenalan ekspresi wajah siswa dengan menggunakan *haar-cascade classifier* dan *neural network*. Metode yang berhasil untuk meraih akurasi sebesar 85% dengan menggunakan dataset JAFFE. Miao, S. et. al. [6] mengusulkan metode pengenalan wajah manusia dengan metode shallow depth CNN yang diberi nama SHCNN. Metode SHCNN berhasil mengklasifikasikan ekspresi wajah manusia dengan akurasi sebesar 69% pada dataset FER-2013. Gan, Y. [7] melakukan review terhadap 4 *pre-trained* CNN model dengan dataset FER-2013. Model tersebut antara lain adalah VGG yang mencapai akurasi tertinggi sebesar 61%, ResNet yang mencapai akurasi tertinggi sebesar 49%, GoogleNet yang mencapai akurasi tertinggi sebesar 64% dan Alex Net yang mencapai akurasi tertinggi sebesar 65%. G.P. Kusuma et al. [8] mengusulkan metode klasifikasi wajah manusia dengan menggunakan VGG-16 + GAP. Metode tersebut berhasil untuk mengklasifikasi ekspresi wajah manusia dengan akurasi sebesar 69.40%.

### III. METODOLOGI

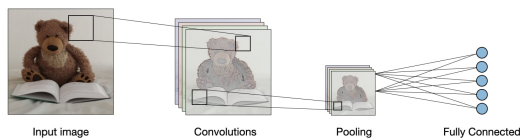
Dalam penelitian ini, metode akan dibagi menjadi 2 bagian yaitu bagian *training* dan *classification*.



Gambar 1. *flowchart training*

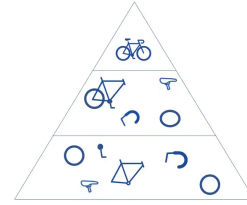
#### A. Training

Bagian *training* merupakan bagian dimana CNN dilakukan training. CNN merupakan *sub-architecture* dari *Neural Network* yang umumnya terdiri atas *convolutional layer*, *pooling layer* dan *fully connected layer*. [9]



Gambar 2. *Convolutional Neural Network* [10]

*Convolutional layer* merupakan *layer* yang menggunakan *filters* untuk melakukan konvolusi kedalam gambar *input*. Hasil dari *convolutional layer* adalah *feature map*. *Non-linearity* di dalam model didapatkan dari penggunaan *ReLU* pada akhir setiap *convolutional layer*. *Convolutional layer* lainnya juga dapat mengikuti *convolutional layer* sebelumnya, dimana jika hal tersebut dilakukan maka akan dibentuk sebuah *feature hierarchy*. [11]



Gambar 3. *Feature Hierarchy* [12]

*Layer* selanjutnya yang terdapat dalam CNN adalah *pooling layer*. *Layer* ini berfungsi untuk mengatasi dimensionalitas dari sebuah *feature map*. *Pooling layer* memiliki cara kerja yang mirip dengan *convolutional layer* namun setiap *filter* yang terdapat pada *pooling layer* tidak memiliki *weights*. *Pooling layer* berguna untuk mengurangi kompleksitas, meningkatkan efisiensi dan mengurangi *overfitting*. [11]

*Layer* terakhir yang terdapat pada CNN adalah *fully connected layer*. *Fully-connected layer* berfungsi untuk melakukan klasifikasi terhadap CNN yang ada. *fully-connected layer* menggunakan fungsi *softmax* untuk menentukan hasil keluaran yang ada. [11]

Arsitektur CNN pada metode yang terdapat pada penelitian ini antara lain adalah sebagai berikut

Tabel 1. Arsitektur CNN

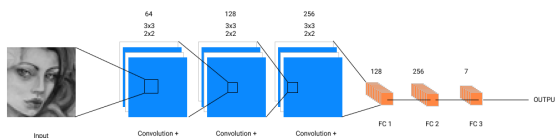
Layer	Ukuran	Jumlah <i>filters</i>
<i>Input Layer</i>	48 x 48	-
<i>Convolution 1</i>	3 x 3	64
<i>Max Pool 1</i>	2 x 2	-
<i>Convolution 2</i>	3 x 3	128
<i>Max Pool 2</i>	2 x 2	-
<i>Convolution 3</i>	3 x 3	256
<i>Max Pool 3</i>	2 x 2	-
<i>Fully Connected 1</i>	-	128
<i>Fully Connected 2</i>	-	256
<i>Fully Connected 3</i>	-	7

Agar tidak terjadi *overfitting* maka pada *Batch Normalization layer* dan *Dropout layer* digunakan di antara *layer-layer* yang terdapat di dalam arsitektur CNN.

*Batch Normalization* adalah teknik yang digunakan untuk standarisasi *input* dari setiap *mini-batch*. *Batch Normalization* memiliki efek untuk menstabilkan proses *learning* dan mengurangi jumlah *training epoch* untuk *men-train* sebuah *neural Network*. [13]

*Dropout* merupakan salah satu teknik *regularization* pada *neural network*. Pada teknik *Dropout*, ketika *training* dijalankan maka beberapa *layer output* akan dibiarkan atau di-*dropped out*. *Dropout* memiliki efek untuk membuat *training* menjadi sedikit *noisy* sehingga membuat *nodes* pada sebuah *layer* untuk memiliki responsibilitas terhadap suatu *input*. [14]

Model CNN yang pada penelitian ini menggunakan optimizer *adam*, dan loss function berupa *sparse categorical loss*.



Gambar 4. Visualisasi Arsitektur CNN

Dataset yang digunakan untuk training pada penelitian ini adalah Open Database FER-2013 [15]. Basis data berisi gambar dengan ukuran 48x48 piksel, yang telah dikategorikan sesuai dengan ekspresi yang ditampilkan. Basis data memiliki 7 jenis ekspresi wajah, yaitu marah, jijik, takut, gembira, sedih, terkejut, dan netral, yang terbagi menjadi dua kategori, yaitu kumpulan data untuk pelatihan dan kumpulan data untuk pengujian. Ada total 28709 gambar untuk pelatihan, dan total 3589 gambar untuk pengujian, total 32.298 gambar untuk diproses.

Sebelumnya, FER-2013 adalah kumpulan data sumber terbuka yang pertama, dibuat untuk proyek yang sedang berjalan oleh Pierre-Luc Carrier dan Aaron Courville, kemudian dibagikan secara publik untuk kompetisi Kaggle, tak lama sebelum ICML 2013.

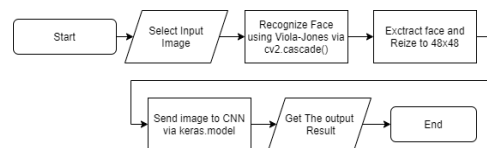
Pada bagian Training kemudian dibagi menjadi beberapa bagian, yang antara lain adalah:

- *Define CNN model*  
Pada tahapan ini arsitektur dari model CNN dibentuk dengan menggunakan library *Tensorflow* [16] dan *Keras* [17] dalam bahasa pemrograman *Python*
- *Prepare Dataset*  
Untuk dapat melakukan training pada CNN maka dibutuhkan dataset. Dataset yang digunakan adalah dataset FER-2013 yang telah dilakukan normalisasi distribusi agar model CNN yang ada tidak terjadi *overfitting*. Normalisasi distribusi dilakukan dengan cara metode *Oversampler*. Kemudian

dataset training kemudian dibagi menjadi 75% data training dan 25% data validasi.

- *Train Data Model*  
Pada tahapan ini, model yang sebelumnya telah dibentuk akan dilakukan training. Training dilakukan dengan dataset berupa dataset yang sebelumnya sudah disiapkan. Pada tahapan ini juga dilakukan konfigurasi *hyperparameter* seperti *epoch*, *batch size*, dan *step* yang digunakan dalam training.
- *Get Accuracy Result*  
Pada tahapan ini, dilakukan evaluasi terhadap model CNN pada metrics akurasi, jika metrics akurasi dirasa sudah mencukupi maka training selesai dan model akan disimpan untuk digunakan pada tahapan selanjutnya, jika akurasi dari model dirasa tidak mencukupi, maka dapat dilakukan modifikasi terhadap *hyperparameter* model agar didapatkan akurasi yang lebih baik.

## B. Classification

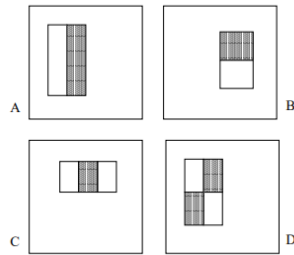


Gambar 5. flowchart Classification

Tahapan *classification* adalah tahapan dimana user dapat mengklasifikasikan dan mendeteksi ekspresi wajah manusia. Hal pertama yang dilakukan untuk mendeteksi ekspresi wajah manusia adalah dengan melakukan deteksi wajah manusia. Algoritma pendeteksi wajah yang kami gunakan adalah *Haar-cascade classifier*.

Dalam melakukan *face recognition* algoritma *Haar-cascade Classifier* melakukan beberapa langkah, antara lain sebagai berikut [18]:

- Memilih *haar-like feature*  
*Haar-like feature* adalah kumpulan fungsi berbentuk persegi, hasil dari *haar-like features*, merupakan angka hasil proses dari fungsi tersebut. *Haar-like features* ini akan digunakan untuk mendeteksi apakah adanya *lines*, *edges* dan juga *diagonal* di dalam *image*, yang membantu untuk melakukan pemisahan *feature* dalam wajah. Dalam algoritma *viola-jones* terdapat 3 *haar-like features*, yaitu *edge*, *lines* dan *diagonal*.



Gambar 6. *Haar-like features* [19]

Berdasarkan gambar diatas, maka gambar A digunakan untuk mendeteksi edges dari wajah, sedangkan gambar B dan gambar C digunakan untuk mendeteksi lines wajah pada subregion gambar, sedangkan gambar D digunakan untuk mendeteksi line diagonal. Haar like feature ini kemudian akan dibagi menjadi sub-window dan akan di iterasi ke seluruh gambar untuk menghasilkan integral image.

- **Membuat Integral image**  
Integral image merupakan sebuah struktur data yang digunakan untuk mempercepat dalam melakukan perhitungan jumlah nilai dari suatu pixel pada suatu gambar. Setiap pixel dari integral image, merupakan hasil penjumlahan dari pixel diatas pixel tersebut, pixel yang terdapat pada bagian kiri pixel tersebut dan juga nilai dari pixel itu sendiri. Untuk menghitung suatu pixel pada gambar kita dapat menggunakan 4 titik untuk perhitungan. Perhitungan keempat titik ini memudahkan dalam perhitungan feature yang didapat dari haar-like features. Dengan menggabungkan haar like feature dengan integral image, kita dapat dengan cepat melakukan perhitungan nilai dari haar-like feature tersebut.
- **Menjalankan Adaboost Training**  
Adaboost merupakan salah satu algoritma machine learning dimana classifier dari algoritma ini merupakan kumpulan-kumpulan dari weak learner yang menghasilkan sebuah strong classifier. Dalam algoritma viola-jones, weak learner yang ada adalah feature-feature yang dihasilkan oleh haar like feature, dimana satu fitur merupakan satu weak classifier. Dalam melakukan face recognition, adaboost melakukan klasifikasi kedalam semua subregion dari gambar dan mengecek apakah classifier yang ada memberikan nilai yang positif atau negatif, jika nilai positif, maka pada gambar tersebut terdapat wajah,

sedangkan jika nilai negatif maka tidak terdapat wajah pada sub-region tersebut.

- **Membuat cascading classifier.**  
Cascading classifier digunakan untuk mengklasifikasi kembali, bagian dari gambar mana yang memiliki wajah dan mana yang tidak merupakan wajah. Cascading classifier digunakan karena hasil dari klasifikasi adaboost masih memiliki jumlah yang cukup banyak, yaitu hampir 8000 features untuk gambar berukuran 24 x 24, dengan feature yang cukup banyak ini, akan membutuhkan waktu yang cukup lama untuk melakukan face recognition, oleh sebab itu cascading classifier dibutuhkan. Cascading classifier bekerja dengan cara membagi menjadi beberapa stage, dimana stage pertama merupakan feature yang dianggap paling bagus untuk merepresentasikan wajah, stage kedua merupakan feature yang lain untuk diklasifikasikan. Dengan menggunakan cascading classifier ini, ketika subregion gambar input yang dimasukan kedalam classifier stage pertama menghasilkan hasil positif, maka subregion gambar tersebut akan diklasifikasikan lagi kedalam stage kedua, namun jika subregion gambar tersebut memiliki nilai yang negatif, maka gambar tersebut akan dibuang dan akan dicari subregion gambar tersebut yang memiliki wajah.

Dalam penelitian ini, *Haar-cascade classifier* diimplementasikan menggunakan library *openCV*. library *openCV* sendiri merupakan library untuk melakukan pengolahan citra yang terdapat pada bahasa pemrograman *python*. Metode *OpenCV* yang digunakan adalah *cv2.cascade()* dimana metode tersebut menerima input berupa file .xml yang berisi hasil training untuk mendeteksi wajah.

Tahapan pada tahap klasifikasi dibagi menjadi beberapa bagian, bagian tersebut antara lain adalah:

- **Select Input Image**  
Pada tahapan ini, user dapat melakukan input image yang berupa gambar wajah manusia ataupun gambar wajah pada *virtual meeting* yang akan dilakukan pengenalan emosi. Input image diterima menggunakan metode *OpenCV* *imread*.
- **Recognize Face**  
Setelah gambar diterima, maka pada gambar tersebut akan dikenali mana yang merupakan wajah manusia dan mana yang bukan. pengenalan ini menggunakan algoritma *haar-cascade* classifier yang telah dijelaskan sebelumnya.



- *Extract Face and Resize to 48x48*  
Setelah wajah terdeteksi, maka akan dilakukan ekstraksi wajah tersebut untuk diteruskan sebagai input pada CNN. Karena input pada CNN memiliki besaran berupa 48x48 pixel, maka gambar wajah yang telah terdeteksi tersebut harus dilakukan perubahan ukuran sehingga gambar menjadi ukuran 48 x 48 pixel.
- *Sent Image to Model via Keras.model*  
Setelah wajah terdeteksi dan diubah menjadi ukuran 48x48 pixel, maka gambar wajah tersebut diberikan menjadi input dalam model CNN menggunakan `keras.predict`. `Keras.predict` digunakan untuk melakukan klasifikasi oleh CNN untuk menghasilkan ekspresi yang sesuai dengan ekspresi yang ada.
- *Get The Output Result*  
Setelah CNN berhasil mengklasifikasikan wajah yang ada, maka ekspresi dapat ditampilkan melalui metode `OpenCV` `imshow`. Wajah manusia yang terdeteksi akan dikelilingi oleh persegi serta teks yang menampilkan ekspresi wajah yang diprediksi.

#### IV. HASIL

Pada penelitian ini, model CNN ditraining menggunakan library *tensorflow-gpu* versi 2.3.0. Tensorflow yang digunakan bukan menggunakan versi 2.4.1 karena pada tensorflow paling terbaru akan terjadi overfitting ketika training dijalankan. Dataset dari training yang digunakan adalah dataset FER-2013 yang didapatkan dari kaggle[]. Dataset FER-2013 ini berbentuk file csv, sehingga diperlukan pre-processing data menggunakan library *pandas*. file csv yang dibaca merupakan file csv dengan nama 'train.csv' Pada data yang telah dibaca, maka akan dilakukan normalisasi terhadap distribusi setiap gambar dari ekspresi wajah dengan menggunakan fungsi `RandomOverSampler` dari library *imblearn*. Hasil dari normalisasi distribusi adalah setiap ekspresi memiliki gambar sebanyak 7215 gambar. Kemudian dari hasil distribusi tersebut, dataset dilakukan pembagian dimana 80% data akan digunakan sebagai data training dan 20% akan digunakan sebagai data test.

Model CNN dibuat pada aplikasi web *google colab* yang menggunakan *Google Compute Backend GPU* dengan *GPU* berupa A100, RAM sebesar 12GB dan *disk storage* sebesar 45GB. *Google Colab* merupakan aplikasi web yang dapat menjalankan file *jupyter notebook* yang dapat menjalankan bahasa pemrograman python.

Model CNN kemudian dibuat menggunakan library *keras* dan *tensorflow* sesuai dengan arsitektur yang diajukan dalam penelitian ini. Hasil dari model CNN pada *tensorflow* dan *keras* antara lain adalah sebagai berikut:

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 64)	640
batch_normalization (Batch Normalization)	(None, 48, 48, 64)	256
max_pooling2d (MaxPooling2D)	(None, 24, 24, 64)	0
dropout (Dropout)	(None, 24, 24, 64)	0
batch_normalization_1 (Batch Normalization)	(None, 24, 24, 64)	256
conv2d_1 (Conv2D)	(None, 24, 24, 128)	73856
batch_normalization_2 (Batch Normalization)	(None, 24, 24, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 128)	0
dropout_1 (Dropout)	(None, 12, 12, 128)	0
batch_normalization_3 (Batch Normalization)	(None, 12, 12, 128)	512
conv2d_2 (Conv2D)	(None, 12, 12, 256)	295168
batch_normalization_4 (Batch Normalization)	(None, 12, 12, 256)	1024
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 256)	0
dropout_2 (Dropout)	(None, 6, 6, 256)	0
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 128)	1179776
batch_normalization_5 (Batch Normalization)	(None, 128)	512
activation (Activation)	(None, 128)	0
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 256)	33024
batch_normalization_6 (Batch Normalization)	(None, 256)	1024
activation_1 (Activation)	(None, 256)	0
dropout_4 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 7)	1799
Total params: 1,588,359		
Trainable params: 1,586,311		
Non-trainable params: 2,048		

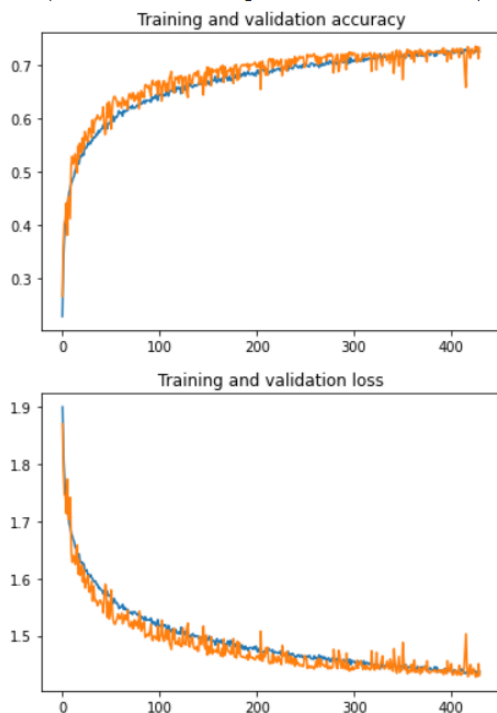
Gambar 7. Hasil dari model.summary()

Model kemudian di-compile dengan parameter berupa *optimizer* yang menggunakan adam, dan *loss* yang menggunakan fungsi *sparse categorical cross entropy*.

Model kemudian dilakukan training dengan menggunakan *hyperparameter* berupa *batch-size* sebesar 64, *steps* dengan rumus jumlah data yang terdapat pada training dibagi dengan 128, *validation split* 0.25 serta *epoch* sebanyak 430. Model juga diberikan fungsi *callback* agar model yang tersimpan adalah model yang memiliki *validation loss* paling sedikit.

Hasil CNN yang di-training sebanyak 430 *epochs* memiliki akurasi sebesar 72.55% pada *training accuracy* dan akurasi sebesar 73.24% pada *validation accuracy*. Model yang disimpan melalui fungsi *callback* memiliki akurasi sebesar 73,15% pada *training accuracy* dan akurasi sebesar 73,56% pada *validation accuracy*.

Hasil grafik kurva dari *accuracy* dan *loss* dari *training* dan *validation loss* dapat dilihat pada gambar 8.



Gambar 8. Grafik Kurva *Training* dan *validation accuracy* dan *loss*, validasi memiliki kurva berwarna jingga dan *training* memiliki kurva berwarna biru

Hasil dari model yang telah di training ketika dijalankan dengan dataset test mencapai akurasi sebesar 73.86% dengan *loss* sebesar 1.42. Hasil *confusion matrix* dari model CNN dengan dataset test dapat dilihat pada gambar 9.

0	9.16%	0.23%	0.69%	0.75%	1.10%	0.24%	1.99%
1	-0.14%	14.11%	0.00%	0.08%	0.00%	0.03%	0.08%
2	-1.12%	0.12%	8.03%	0.76%	1.34%	1.47%	1.68%
3	-0.39%	0.05%	0.29%	11.47%	0.71%	0.42%	1.37%
4	-0.97%	0.14%	1.11%	0.50%	8.24%	0.26%	2.72%
5	-0.26%	0.07%	0.45%	0.47%	0.28%	12.24%	0.59%
6	-0.60%	0.02%	0.35%	0.89%	1.33%	0.12%	10.62%
	0	1	2	3	4	5	6

Gambar 9. Hasil *confusion matrix*. Sumbu X menunjukkan hasil asli dari test dan sumbu Y merupakan hasil prediksi model

Pada *confusion matrix*, label 0 - 6 mendandakan setiap ekspresi yang ada dimana label 0 merupakan ekspresi senang, label 1 merupakan ekspresi jijik, label 2 merupakan ekspresi takut, label 3 merupakan ekspresi senang, label 4 merupakan ekspresi sedih, label 5 merupakan ekspresi kaget dan label 6 merupakan ekspresi netral.

Contoh hasil model CNN pada dataset test dapat dilihat pada gambar 10.



Result: Fear True: Fear



Result: Disgust True: Disgust



Result: Angry True: Angry

Gambar 10. Visualisasi Hasil dari test

Kemudian untuk bagian *classification*, menggunakan library *OpenCV* yang digunakan untuk membaca file gambar, melakukan *preprocessing* pada gambar. kemudian untuk menaruh teks pada gambar.

Bagian *classification* dijalankan pada *jupyter notebook* yang menjalankan *tensorflow* versi 2.4.1 serta berjalan pada perangkat keras dengan spesifikasi sebagai berikut:

CPU : Intel i7-8750H

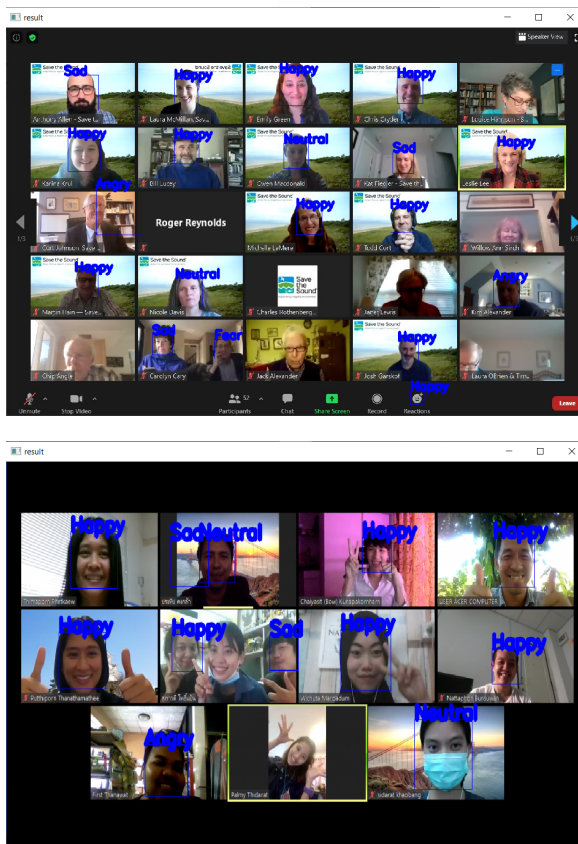
GPU: Nvidia GTX1050 4GB

RAM: 8 GB

kemudian untuk melakukan *face recognition* digunakan fungsi *cv2.CascadeClassifier* dan *cv2.detectMultiScale*. *Cascade classifier* yang digunakan didapatkan dari Github *OpenCV* dengan nama 'haarcascade\_frontalface\_default.xml'. Model CNN dimasukan dengan cara menggunakan fungsi *keras.model.load\_model* dengan parameter model yang telah di train sebelumnya. kemudian untuk melakukan pengenalan ekspresi maka digunakan fungsi *model.predict* yang menerima hasil gambar 48x48 wajah dari gambar yang tadi telah dimasukan.

Hasil dari pengenalan ekspresi muka manusia dapat dilihat pada gambar dengan menggunakan data

screenshot dari *virtual meeting environment* yang berupa *zoom meeting* yang diambil dari internet.



Gambar 11. Pengenalan ekspresi wajah manusia pada *virtual meeting environment* [19][20]

## V. ANALISIS

Berdasarkan dari hasil yang didapatkan, maka dapat disimpulkan bahwa model memiliki hasil yang cukup baik dalam melakukan pengenalan terhadap ekspresi wajah manusia. Hal ini didukung dengan akurasi yang cukup tinggi yaitu 73%. Grafik kurva dari *training* dan *validation* juga menunjukkan bahwa model CNN yang terdapat pada penelitian tidak terjadi *overfitting*.

*Epoch* sebesar 430 dipilih karena jika *epoch* melebihi 450 maka akan terjadi *overfitting* dan kemudian jika *epoch* lebih kecil dari 430, maka akurasi yang didapatkan akan menjadi tidak akurat. *Batch size* sebesar 64 digunakan sebagai *batch size* yang cukup untuk *training* dengan GPU dan agar tidak terjadi *convergence* yang lebih cepat sehingga menghindari terjadinya *overfitting*.

Pada *confusion matrix* model dapat dilihat bahwa 3 nilai error terbesar terdapat value *neutral* dengan *sad*, *neutral* dengan *happy* dan *neutral* dengan *fear*. Hal ini dapat terjadi karena dataset FER2013 memiliki gambar yang cukup mirip antara ekspresi *sad*, *neutral* dan *happy*. Sehingga ketika melakukan klasifikasi maka akan terjadi kesalahan dalam pengenalan ekspresi tersebut.

Pada percobaan dengan data asli, dapat dilihat bahwa terdapat beberapa wajah yang tidak terdeteksi. Hal tersebut dapat terjadi karena algoritma *haar-cascade* tidak dapat mendeteksi ketika pencahayaan kurang, serta ketika wajah orang tidak berupa wajah *front-frontal*.

Pada *virtual meeting environment* juga terdapat kelemahan, seperti tipe kamera yang berbeda-beda setiap orang sehingga resolusi dan pencahayaan yang dihasilkan juga cukup buruk untuk algoritma *haar-cascade* untuk melakukan *face-recognition*. Kecepatan Internet juga berpengaruh terhadap resolusi gambar dari wajah manusia yang mengikuti *virtual meeting* sehingga menyebabkan dapat terjadinya kesalahan pada saat pengenalan ekspresi dilakukan.

## VI. SIMPULAN

Dapat disimpulkan bahwa proyek dapat berjalan dengan baik meskipun ditemui beberapa kendala selama proses pengujian, namun kendala tersebut dapat dijadikan sebagai Batasan yang tentunya kendala tersebut didukung oleh argumen dan oleh karena itu dapat menginspirasi umpan balik untuk penelitian selanjutnya.

Diharapkan melalui pendeteksian wajah ini, pendeteksi wajah ini dapat digunakan untuk mengetahui kepuasan seseorang dalam berkomunikasi, membantu menentukan karakteristik seseorang, dan mendapatkan masukan nonverbal dari lawan komunikasi yang kita ajak bicara.

Hasil dari proyek ini diharapkan dapat direalisasikan agar dapat digunakan umpan balik dalam sistem pembelajaran atau pembelajaran online untuk mengetahui tingkat kepuasan dan konsentrasi mahasiswa atau mahasiswa dalam perkuliahan atau ruang kelas online, sebagai sistem umpan balik ketika melakukan tes pasar sistematis dan untuk membantu menemukan masalah psikologis seseorang.

## UCAPAN TERIMA KASIH

Dengan demikian, kami ingin mengucapkan terimakasih sebesar-besarnya kepada Ibu Nabila Husna Shabrina, S.T., M.T. selaku dosen pengajar mata kuliah Pemrograman Visual, yang telah membimbing kami dalam pengerjaan kajian ini.

Kami juga berterima kasih kepada pembaca yang telah mengulas kajian ini, dan kami harap dapat memberikan manfaat bagi pembaca.

## DAFTAR PUSTAKA

- [1] U. F. R. Rahmawaty and E. M. Lokollo, "WORK FROM HOME," ISED, 2020.
- [2] J. A. Devito, in *Interpersonal communication book*, global edition, Pearson Education Limited, 2015, p. 139.
- [3] I. Lasri, A. R. Solh, and M. E. Belkacemi, "Facial Emotion Recognition of Students using Convolutional Neural Network," 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), 2019.

- [4] T. Connie, M. Al-Shabi, W. P. Cheah, and M. Goh, "Facial Expression Recognition Using a Hybrid CNN-SIFT Aggregator," *Lecture Notes in Computer Science*, pp. 139–149, 2017.
- [5] D. Yang, A. Alsadoon, P. W. C. Prasad, A. K. Singh, and A. Elchouemi, "An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment," *Procedia Comput. Sci.*, vol. 125, no. 2009, pp. 2–10, 2018, doi:10.1016/j.procs.2017.12.003.
- [6] S. Miao, H. Xu, Z. Han, and Y. Zhu, "Recognizing Facial Expressions Using a Shallow Convolutional Neural Network," *IEEE Access*, vol. 7, pp. 78000–78011, 2019.
- [7] Y. Gan, "Facial Expression Recognition Using Convolutional Neural Network," *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, 2018.
- [8] G. P. Kusuma, J. Jonathan, and A. P. Lim, "Emotion Recognition on FER-2013 Face Images Using Fine-Tuned VGG-16," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 6, pp. 315–322, 2020.
- [9] "Convolutional Neural Networks cheatsheet," CS 230 - Convolutional Neural Networks Cheatsheet. [Online]. Tersedia: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks>. [Diakses: 16-Mei-2021].
- [10] S. Amidi and A. Amidi, *architecture-cnn-en.jpeg*. Stanford University.
- [11] IBM Cloud Education, "What are Convolutional Neural Networks?," IBM. [Online]. Tersedia: <https://www.ibm.com/cloud/learn/convolutional-neural-networks>. [Diakses: 16-Mei-2021].
- [12] IBM Cloud Education, *Feature Hierarchy.jpg*.
- [13] J. Brownlee, "A Gentle Introduction to Batch Normalization for Deep Neural Networks," *Machine Learning Mastery*, 03-Dec-2019. [Online]. Tersedia: <https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/>. [Diakses: 18-Mei-2021].
- [14] J. Brownlee, "A Gentle Introduction to Dropout for Regularizing Deep Neural Networks," *Machine Learning Mastery*, 06-Aug-2019. [Online]. Tersedia: <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>. [Diakses: 18-Mei-2021].
- [15] "Challenges in Representation Learning: Facial Expression Recognition Challenge," Kaggle. [Online]. Tersedia: <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data?select=fer2013.tar.gz>. [Diakses: 10-Mei-2021].
- [16] TensorFlow. [Online]. Tersedia: <https://www.tensorflow.org/>. [Diakses: 18-May-2021].
- [17] K. Team, "Simple. Flexible. Powerful.," Keras. [Online]. Tersedia: <https://keras.io/>. [Diakses: 18-May-2021].
- [18] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*.
- [19] *Save The Sound, image.jpg*. Save The Sound.
- [20] P. Kongkla, *Screenshot-18.png*. WALAILAK UNIVERSITY.