

# Deteksi Ekspresi Wajah Manusia Menggunakan Metode *Convolutional Neural Network*

Christian Cahyaningtyas<sup>1</sup>, Mira<sup>2</sup>, Candra Gudiato<sup>3</sup>, Maya Sari<sup>4</sup>

<sup>1,2,3,4</sup> Teknologi Informasi, Institut Shanti Bhuana

<sup>1</sup>christi@shantibhuana.ac.id\*, <sup>2</sup>mira@shantibhuana.ac.id, <sup>3</sup>candra.gudiato@shantibhuana.ac.id,

<sup>4</sup>maya.sari@shantibhuana.ac.id

## Abstract

*Detecting emotions based on human facial expressions has long been a focus of research in the field of psychology that identifies the relationship between expressions and a person's emotional state. Deep Learning has the ability to learn data representations with a high level of abstraction, allowing models to make predictions or decisions based on big data without human intervention and Deep Learning can also understand and classify an object. One of the main breakthroughs in the field of Deep Learning is the Convolutional Neural Network, its advantage is that it has very good performance results in various tasks in image processing. In this study, research will be conducted on detecting human facial expressions using the Convolutional Neural Network method by conducting several experiments. Image data will be processed using the python programming language and the dataset used is data sourced from the official kaggle.com website, namely FER2013. Based on the results of the Convolutional Neural Network model training for facial expression recognition, the highest accuracy reached 60% at the 60th epoch with batch sizes of 64 and 128. When viewed based on the graph produced, the best model training was carried out with a batch size of 128 epoch 60. This is because validation stability is achieved faster, less fluctuating and the model is not overfitting and the model learns better on new data. When viewed from the Confusion Matrix produced by batch size 64 epoch 60, the resulting model is more balanced in detecting human facial emotions. While from the classification report produced, there is no significant difference. In general, the performance produced is almost similar. However, the batch size of 64 epoch 60 is slightly better.*

*Keywords: detection, face ekspretion, convolutional neural network*

## Abstrak

Deteksi emosi berdasarkan ekspresi wajah manusia sudah menjadi fenomena yang sudah lama menjadi fokus penelitian dalam bidang psikologis yang mengidentifikasi hubungan antara ekspresi dan kondisi emosi seseorang. *Deep Learning* memiliki kemampuan untuk belajar representasi data dengan tingkat abstraksi yang tinggi, memungkinkan model untuk membuat prediksi atau keputusan berdasarkan data besar tanpa intervensi manusia serta *Deep Learning* juga bisa memahami dan mengklasifikasi suatu objek. Salah satu terobosan utama dalam bidang *Deep Learning* adalah *Convolutional Neural Network*, keunggulannya adalah mempunyai hasil kinerja yang sangat baik dalam berbagai tugas dalam pengolahan citra. Pada penelitian ini akan dilakukan penelitian tentang deteksi ekspresi wajah manusia menggunakan metode *Convolutional Neural Network* dengan melakukan beberapa percobaan. Data gambar akan diolah menggunakan bahasa pemrograman *python* dan dataset yang digunakan merupakan data bersumber dari website resmi *kaggle.com* yaitu FER2013. Berdasarkan hasil pelatihan model *Convolutional Neural Network* untuk pengenalan ekspresi wajah, akurasi tertinggi mencapai 60% pada *epoch* ke-60 dengan *batch size* 64 dan 128. Jika dilihat berdasarkan dari grafik yang dihasilkan pelatihan model terbaik dilakukan dengan *batch size* 128 *epoch* 60. Hal tersebut dikarenakan stabilitas validasi lebih cepat tercapai, lebih sedikit fluktuatif dan model tidak *overfitting* serta model belajar lebih baik terhadap data yang baru. Jika dilihat dari *confusion matrix* yang dihasilkan *batch size* 64 *epoch* 60 model yang dihasilkan lebih seimbang dalam mendeteksi emosi wajah manusia. Sedangkan dari *classification report* yang dihasilkan tidak ada perbedaan yang signifikan. Secara umum performa yang dihasilkan hampir mirip. Namun *batch size* 64 *epoch* 60 sedikit lebih baik.

Kata kunci: deteksi, ekspresi wajah, *convolutional neural network*

©This work is licensed under a Creative Commons Attribution -ShareAlike 4.0International License

## 1. Pendahuluan

Wajah adalah objek yang digunakan untuk mengidentifikasi seseorang, karena dari paras wajah manusia dapat terlihat emosi atau ekspresi yang muncul dalam berinteraksi sosial. Ekspresi atau paras wajah merupakan bentuk komunikasi nonverbal dimana ekspresi tersebut menunjukkan keadaan emosi seseorang [1-2]. Dengan mengetahui keadaan emosi seseorang pada kondisi tertentu dapat mengetahui kondisi psikologi seseorang. Emosi yang dihasilkan

dapat mempengaruhi gerak otot wajah yang dapat diidentifikasi sebagai ekspresi wajah seseorang. Gerakan otot pada wajah seseorang dapat berbeda-beda tergantung dengan emosi yang sedang dirasakan oleh seseorang. Dengan memahami kondisi emosional setiap individu dalam waktu dan situasi tertentu maka dapat mengevaluasi tingkat kesejahteraan psikologis mereka [3]. Ekspresi wajah manusia dapat diklasifikasikan menjadi tujuh, yaitu senang, sedih, takut, jijik, marah, terkejut dan biasa saja [4].

Deteksi wajah adalah salah satu hal yang paling kompleks dan yang menantang dalam bidang visual komputer, karena sangat banyak variasi yang disebabkan oleh perubahan seperti penampilan wajah, pencahayaan dan ekspresi wajah manusia [5]. Deteksi emosi berdasarkan ekspresi wajah manusia sudah menjadi fenomena yang sudah lama menjadi fokus penelitian dalam bidang psikologis yang mengidentifikasi hubungan antara ekspresi dan kondisi emosi seseorang. Memahami dan mendeteksi emosi berdasarkan ekspresi wajah manusia mempunyai implikasi yang sangat signifikan dalam beberapa konteks, termasuk dalam berinteraksi sosial, komunikasi, dan memahami emosi seseorang [6].

Dengan adanya kemajuan teknologi kecerdasan buatan dan pembelajaran mesin, sistem deteksi wajah mampu mengenali berbagai ekspresi, sudut pandang, dan pencahayaan dengan lebih akurat dan efisien. Seiring dengan perkembangan ini, deteksi wajah yang dikombinasikan dengan metode pengenalan dapat dimanfaatkan sebagai dasar untuk mengidentifikasi ekspresi wajah manusia [7]. Salah satu pendekatan yang bisa digunakan adalah algoritma *Deep Learning*, yang telah menunjukkan hasil yang luar biasa dalam bidang *computer vision*. *Deep Learning* memiliki kemampuan untuk belajar representasi data dengan tingkat abstraksi yang tinggi, memungkinkan model untuk membuat prediksi atau keputusan berdasarkan data besar tanpa intervensi manusia serta *Deep Learning* juga bisa memahami dan mengklasifikasi suatu objek [8].

Salah satu terobosan utama dalam bidang *Deep Learning* adalah *Convolutional Neural Network (CNN)*, yaitu sebuah model atau metode yang banyak digunakan dalam memproses data visual seperti gambar dan telah terbukti sangat efektif dalam berbagai aplikasi di bidang *computer vision* [9]. *Convolutional Neural Network* merupakan salah satu metode dalam klasifikasi yang menggunakan layer konvolusi untuk mengonvolusi suatu *input* dengan *filter*. Keunggulan dari *Convolutional Neural Network* adalah mempunyai hasil kinerja yang sangat baik dalam berbagai tugas dalam pengolahan citra [10-11]. Karena *Convolutional Neural Network* mampu mengklasifikasikan suatu gambar atau objek dan menghasilkan akurasi yang sangat baik [12-13].

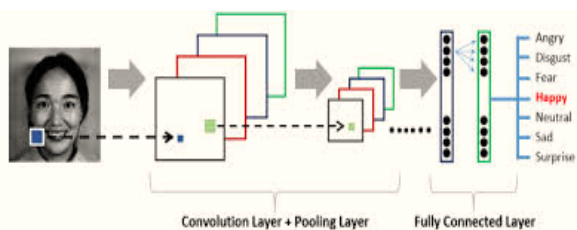
Adapun hasil penelitian lainnya yang sudah pernah dilakukan terkait dengan menggunakan metode *Convolutional Neural Network* yaitu untuk membantu penderita autisme untuk mengenali ekspresi wajah seseorang, dimana tujuannya dari penelitian ini adalah untuk membantu penderita autisme dalam bersosialisasi dengan orang lain. Luaran yang dihasilkan dari penelitian ini berupa tampilan hasil deteksi ekspresi wajah berupa diagram dan keterangan ekspresi yang sedang dialami oleh seseorang [14]. Penelitian lainnya tentang klasifikasi ekspresi wajah manusia menggunakan model arsitektur *Xception* pada metode *Convolutional Neural Network* dengan

menggunakan dataset FER2013. Aplikasi tersebut dirancang dengan menggunakan *library OpenCV*, *TensorFlow* dan *Keras* sebagai *backend*. Adapun *output* yang dihasilkan dari penelitian tersebut adalah akurasi sebesar 66%, presisi 76%, recall 65% dan score F1 63% [15]. Penelitian lain mengenai proses pengenalan ekspresi wajah dengan menerapkan metode *Convolutional Neural Network* yang berfokus pada pengembangan model *Convolutional Neural Network* dengan *library TensorFlow* dengan melakukan uji coba sebanyak tiga kali menghasilkan *training accuracy* tertinggi sebesar 71% dan *validation accuracy* sebesar 65%. Selain itu, pada penelitian tersebut peneliti juga mengimplementasikan model yang dibuat pada website dengan *video realtime*. Sistem dibangun dengan menggunakan *framework Django* dan terdapat beberapa *library python* yang digunakan [4]. Penelitian dengan mengimplementasikan *deep learning* dengan menggunakan metode *Convolutional Neural Network* pada ekspresi manusia juga sudah pernah dilakukan. Dengan menggunakan bantuan bahasa pemrograman *python*, web berbasis *flask*, *Tensorflow* dan *Opencv* dalam proses pengaplikasiannya. Pada penelitian tersebut proses training dilakukan dengan menggunakan *batch size* dan *epoch* yang berbeda untuk menghasilkan model yang baik. Kesimpulan dari penelitian tersebut adalah dengan *epoch 100 batch size 128* didapat *training accuracy* sebesar 90% dan *validation 65%* serta percobaan yang dilakukan sebanyak 35 ekspresi, yang berhasil ditebak sebanyak 28 ekspresi dengan tingkat akurasi sebesar 80% [16]. Ekspresi wajah seorang anak usia dini secara tidak langsung juga memperlihatkan isi dari pikiran anak yang dapat menjadi suatu kebiasaan bagi anak sehingga dapat berpengaruh pada tumbuh kembang dan proses pertumbuhan anak pada usia dini. Penelitian dengan menggunakan model *face expression recognition* dapat digunakan dalam mengidentifikasi ekspresi wajah anak usia dini. Penelitian tersebut menerapkan metode *Convolutional Neural Network*, memanfaatkan *library opencv* pada *python* dan *framework Django*. Hasil yang didapat dari penelitian tersebut adalah berupa aplikasi berbasis website yang dapat digunakan secara *realtime*, serta tampilan yang dihasilkan dari aplikasi tersebut berupa ekspresi dari masing-masing wajah dan persentase jenis ekspresi yang didapat [17].

Berdasarkan penjelasan sebelumnya, metode *Convolutional Neural Network* dapat digunakan dalam mendeteksi ekspresi wajah manusia. Tujuan dari penelitian ini adalah untuk mengimplementasikan metode *Convolutional Neural Network* dalam mendeteksi ekspresi wajah manusia dengan melakukan beberapa percobaan pada saat pelatihan model dengan *batch size* dan *epoch* yang berbeda-beda untuk mendapatkan hasil akurasi terbaik. Pengolahan data gambar akan dilakukan menggunakan bahasa pemrograman *python* dan dataset yang akan digunakan merupakan data yang bersumber dari situs resmi *kaggle* yaitu FER2013.

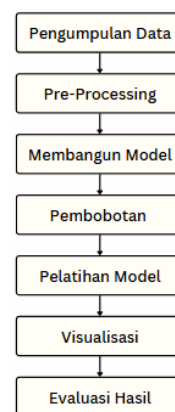
## 2. Metode Penelitian

Penelitian ini menggunakan metode *Convolutional Neural Network* untuk mendeteksi ekspresi wajah manusia. *Convolutional Neural Network* merupakan jaringan saraf yang mempunyai tiga *layer*, yaitu *convolutional layer*, *pooling layer* dan *fully connected layer* [18-19]. *Convolutional layer* merupakan tahap awal dalam pemrosesan citra, yang digunakan untuk mengekstrak fitur dari inputan. Untuk mendeteksi pola seperti tepi, tekstur dan bentuk dapat menggunakan *filter* yang akan bergerak melintasi gambar [20-22]. Dari proses tersebut menghasilkan *feature map* dan diaktivasi dengan menggunakan fungsi non linear seperti ReLU. Untuk mengurangi dimensi pada *feature map* agar dapat mempercepat proses komputasi maka dilakukan *pooling layer* [23]. *Pooling layer* sendiri terdiri dari sebuah *filter* dengan ukuran tertentu yang dioperasikan menggunakan *stride* tertentu pada *feature map*. Proses ini dilakukan dengan cara menggeser *filter* keseluruhan area *feature map* dan menerapkan fungsi seperti *max pooling* atau *average pooling* pada setiap area yang dicakup oleh *filter* tersebut. Sedangkan *fully connected layer* yaitu menghubungkan neuron yang serupa dengan tujuan untuk representasi data agar data dapat diklasifikasikan secara linear [24]. Serta biasanya pada lapisan ini menggunakan fitur aktivasi *softmax* untuk menghasilkan probabilitas klasifikasi. Namun sebelum melakukan klasifikasi, *feature map* hasil dari ekstraksi fitur yang masih berbentuk *array* multidimensi diproses menggunakan teknik *Global Average Pooling*. Teknik ini bekerja dengan menghitung rata-rata nilai dari setiap *channel* pada *feature map*, sehingga menghasilkan vektor satu dimensi yang lebih ringkas. Vektor hasil *Global Average Pooling* ini kemudian digunakan sebagai *input* untuk *fully connected layer*, yang selanjutnya memproses dan mengklasifikasikan data sesuai dengan jumlah kelas yang ada [25]. Gambar 1 merupakan gambar dari arsitektur *Convolutional Neural Network*.



Gambar 1. Arsitektur *Convolutional Neural Network*.

Proses pengolahan data menggunakan aplikasi *google collaboration* dan beberapa *library* pada *python*. Adapun alur kerja atau proses dalam penelitian ini terdapat beberapa tahapan yang akan dilakukan, yaitu dimulai dengan tahap pengumpulan data, tahap *preprocessing* data, tahap membangun model, tahapan pembobotan, tahapan pelatihan model, tahapan visualisasi dan evaluasi. Adapun gambar dari tahapan penelitian dapat dilihat pada Gambar 2.



Gambar 2. Tahapan Penelitian

Penjelasan dari Gambar 2 pada tahapan penelitian yang dilakukan adalah sebagai berikut :

### 2.1 Pengumpulan data

Data yang digunakan dalam penelitian ini diambil dari website terpercaya atau resmi yaitu *kaggle.com*. Dataset yang diambil merupakan data mengenai gambar dari ekspresi wajah manusia, dataset yang digunakan yaitu dataset FER2013. Gambar ekspresi wajah manusia yang digunakan dalam bentuk file *.png*. Data tersebut dibagi menjadi tujuh kelas, yaitu *happy*, *angry*, *disgust*, *fear*, *neutral*, *sad* dan *surprise* dengan jumlah keseluruhan terdapat 35.887 data ekspresi wajah manusia.

### 2.2 Pre-processing

Sebelum data siap untuk digunakan, dataset tersebut harus dilakukan proses *pre-processing* terlebih dahulu. Pada tahap *pre-processing* dalam penelitian ini dilakukan proses yang dimulai dengan melakukan normalisasi gambar dengan menggunakan *rescaling*. Selanjutnya dilakukan augmentasi data pada data *training* untuk meningkatkan keragaman data. Setelah itu tentukan *target size*, *color mode*, *class mode*, *batch size* dan *shuffle* sesuai dengan kebutuhan. Pada tahap *pre-processing* ini tujuannya untuk menyiapkan data dalam format yang sesuai agar model dapat dilatih secara efektif dan validasi dapat dilakukan secara optimal.

### 2.3 Membangun Model

Setelah dilakukan tahap *pre-processing* dan data sudah siap digunakan maka dilanjutkan dengan membangun model dengan menggunakan metode *Convolutional Neural Network*. Model *Convolutional Neural Network* dibangun dengan menggunakan tiga lapisan utama, yaitu *convolutional layer*, *pooling layer* dan *fully connected layer*. Pada *convolutional layer* digunakan untuk mengekstraksi fitur gambar. *Pooling layer* untuk mengurangi ukuran data dan mempercepat proses. Setelah itu, hasilnya diratakan menggunakan *GlobalAveragePooling* dan diproses oleh *fully connected layer* untuk menghasilkan prediksi akhir tujuh kelas ekspresi melalui aktivasi *softmax*. Serta pada pembangunan model ini regularisasi seperti *batch*

*normalization* dan *dropout* digunakan untuk menstabilkan pelatihan dan mencegah terjadinya *overfitting*.

#### 2.4 Pembobotan

Pada pembobotan ini dilakukan perhitungan bobot kelas dengan menggunakan fungsi *compute class weight* dari *library scikit-learn* untuk mengatasi ketidakseimbangan jumlah data pada masing-masing kelas. Pemberian bobot dilakukan secara otomatis berdasarkan dengan jumlah data tiap kelas. Pemberian bobot ini bertujuan untuk mencegah model menjadi bias terhadap kelas mayoritas serta memastikan bahwa kelas minoritas tetap diperhatikan selama proses pelatihan.

#### 2.5 Pelatihan model

Pada tahap pelatihan model ini dilakukan beberapa kali percobaan dengan *epoch* dan *batch size* yang berbeda untuk mendapatkan hasil yang optimal. Pada pelatihan ini menggunakan *batch size* 64, 128, dan 256 serta *epoch* 30 dan 60. Pada pelatihan model ini juga dilakukan pengaturan menggunakan *callbacks* yang berfungsi untuk membantu dalam proses training model agar mendapatkan hasil yang optimal dan lebih efisien. *Callbacks* yang digunakan dalam pelatihan model ini ada dua yaitu *EarlyStopping* dan *ReduceLROnPlateau*. *EarlyStopping* digunakan untuk menghentikan *training* lebih awal jika model tidak memberikan hasil yang lebih baik. Sehingga hal ini dapat mencegah terjadinya *overfitting* dan dapat menghemat waktu. Sedangkan *ReduceLROnPlateau* digunakan untuk menurunkan *learning rate* saat model stagnan, hal ini dilakukan agar proses *learning* menjadi lebih baik dan model dapat mencapai performa yang lebih baik lagi.

#### 2.6 Visualisasi

Setelah data dilatih maka selanjutnya visualisasi hasil kinerja dari model untuk melihat seberapa baik model melakukan deteksi. Visualisasi hasil ini berupa grafik akurasi, *confusion matrix* dan *classification report*. Grafik akurasi ini akan menunjukkan perkembangan nilai dari akurasi model selama proses training berlangsung dari *epoch* ke *epoch* berikutnya. *Confusion matrix* digunakan untuk menggambarkan jumlah prediksi yang benar dan salah pada setiap kelasnya. Sehingga dapat dilihat secara rinci pada kelas mana model paling sering salah dalam memprediksi dan seberapa baik model membedakan antar kelas. Sedangkan *classification report* dapat memberikan metrik evaluasi lebih detail untuk setiap kelas, seperti *precision*, *recall*, *f1-score* dan *support*. Hasil dari visualisasi kinerja tersebut selanjutnya dapat dijadikan bahan analisis lebih lanjut.

#### 2.7 Evaluasi

Tahap yang terakhir adalah evaluasi, dimana dilakukan penilaian kinerja model berdasarkan grafik akurasi, *confusion matriks* dan *classification report*.

Berdasarkan grafik akurasi dapat dilihat apakah ada kestabilan dan kecenderungan terjadinya *overfitting* atau *underfitting*. *Confusion matrix* menunjukkan ketepatan dalam memprediksi kelas, sedangkan *classification report* memberikan hasil *precision*, *recall* dan *f1-score*. Dari hasil evaluasi ini dapat digunakan untuk menyimpulkan apakah model sudah optimal atau masih perlu adanya perbaikan.

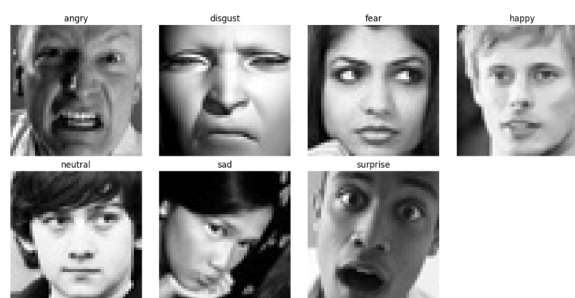
### 3. Hasil dan Pembahasan

Pada penelitian ini dataset yang digunakan merupakan data yang bersumber dari website resmi atau terpercaya yaitu *kaggle.com* dengan alamat *link* sebagai berikut : <https://www.kaggle.com/datasets/msambare/fer2013> [26]. Dataset yang digunakan merupakan data ekspresi wajah manusia yang terdiri dari ekspresi tujuh ekspresi yaitu *happy*, *sad*, *disgust*, *angry*, *neutral*, *fear* dan *surprise*. Dataset tersebut dibagi menjadi dua bagian yaitu data *training* dan data *testing* dengan jumlah 28.709 data untuk data *training* dan 7.178 data *testing*. Adapun jumlah ekspresi wajah berdasarkan dengan masing-masing ekspresinya dapat dilihat pada Tabel 1.

Tabel 1. Jumlah Dataset

| Ekspresi        | Training | Testing |
|-----------------|----------|---------|
| <i>Happy</i>    | 7.215    | 1.774   |
| <i>Fear</i>     | 4.097    | 1.024   |
| <i>Sad</i>      | 4.830    | 1.247   |
| <i>Neutral</i>  | 4.965    | 1.233   |
| <i>Angry</i>    | 3.995    | 958     |
| <i>Disgust</i>  | 436      | 111     |
| <i>Surprise</i> | 3.171    | 831     |

Contoh gambar atau data yang digunakan dalam penelitian ini dapat dilihat dengan jelas pada Gambar 3. Pada gambar terlihat terdapat tujuh ekspresi wajah manusia yang digunakan. Ekspresi tersebut diantaranya yaitu ekspresi *angry*, *disgust*, *fear*, *happy*, *neutral*, *sad* dan *surprise*.



Gambar 3. Ekspresi Wajah Manusia.

Dataset tersebut sebelum dimodelkan harus dilakukan *pre-processing* data terlebih dahulu agar data tersebut dalam format yang dibutuhkan dan model dapat dilatih secara efektif dan validasi dapat dilakukan secara optimal. Setelah data siap untuk dimodelkan maka data tersebut dimodelkan dengan menggunakan metode

Convolutional Neural Network dan pada Gambar 4 merupakan arsitektur yang dihasilkan.

| Layer (type)  | Output Shape        | Param # |
|---|---------------------|---------|
| input_layer_5 (InputLayer)                          | (None, 48, 48, 3)   | 0       |
| conv2d_15 (Conv2D)                                  | (None, 48, 48, 64)  | 608     |
| batch_normalization_15 (BatchNormalization)         | (None, 48, 48, 64)  | 256     |
| max_pooling2d_15 (MaxPooling2D)                     | (None, 24, 24, 64)  | 0       |
| dropout_20 (Dropout)                                | (None, 24, 24, 64)  | 0       |
| conv2d_16 (Conv2D)                                  | (None, 24, 24, 128) | 73,856  |
| batch_normalization_16 (BatchNormalization)         | (None, 24, 24, 128) | 512     |
| max_pooling2d_16 (MaxPooling2D)                     | (None, 12, 12, 128) | 0       |
| dropout_21 (Dropout)                                | (None, 12, 12, 128) | 0       |
| conv2d_17 (Conv2D)                                  | (None, 12, 12, 256) | 295,168 |
| batch_normalization_17 (BatchNormalization)         | (None, 12, 12, 256) | 1,824   |
| max_pooling2d_17 (MaxPooling2D)                     | (None, 6, 6, 256)   | 0       |
| dropout_22 (Dropout)                                | (None, 6, 6, 256)   | 0       |
| global_average_pooling2d_5 (GlobalAveragePooling2D) | (None, 256)         | 0       |
| dense_18 (Dense)                                    | (None, 512)         | 131,328 |
| dropout_23 (Dropout)                                | (None, 512)         | 0       |
| dense_11 (Dense)                                    | (None, 7)           | 3,581   |

Gambar 4. Arsitektur Convolutional Neural Network

Dataset yang telah dimodelkan selanjutnya diberikan bobot terlebih dahulu pada setiap kelasnya sebelum dilakukan pelatihan model. Hal ini dilakukan karena terdapat ketidakseimbangan data. Jika terdapat data yang tidak seimbang maka hasil yang diberikan juga tidak akan optimal dan akurat. Pada penelitian ini perhitungan pembobotan dilakukan secara otomatis sebelum dilakukan *training* model. Berikut merupakan bobot yang di berikan pada setiap kelasnya yang dapat dilihat pada Tabel 2.

Tabel 2. Pembobotan Pada Setiap Kelas

| Ekspresi        | Bobot |
|-----------------|-------|
| <i>Happy</i>    | 0,57  |
| <i>Fear</i>     | 1,00  |
| <i>Sad</i>      | 0,85  |
| <i>Neutral</i>  | 0,83  |
| <i>Angry</i>    | 1,03  |
| <i>Disgust</i>  | 9,41  |
| <i>Surprise</i> | 1,29  |

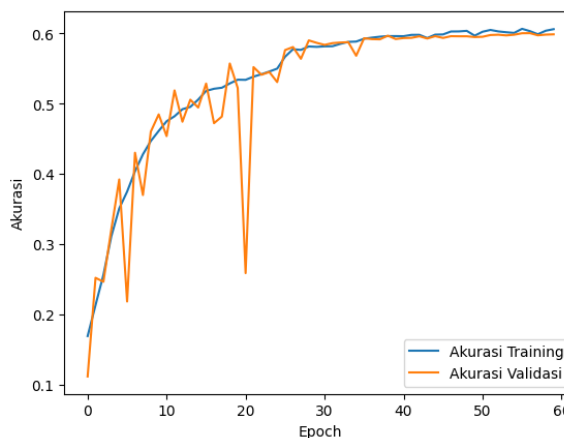
Berdasarkan Tabel 2, pemberian bobot dilakukan dengan cara, jumlah data yang sedikit akan diberi bobot yang tinggi. Sedangkan jika jumlah data pada kelas berjumlah banyak maka diberikan bobot rendah. Jadi semakin banyak datanya maka bobot yang diberikan semakin kecil. Hal ini dilakukan dengan tujuan agar data menjadi lebih seimbang. Setelah pembobotan selesai dilakukan maka dilanjutkan ke proses pelatihan model.

Pada proses pelatihan model, dilakukan beberapa kali percobaan dengan *batch size* dan *epoch* yang berbeda untuk mencari hasil yang terbaik. Adapun hasil dari percobaan tersebut dapat dilihat pada Tabel 3.

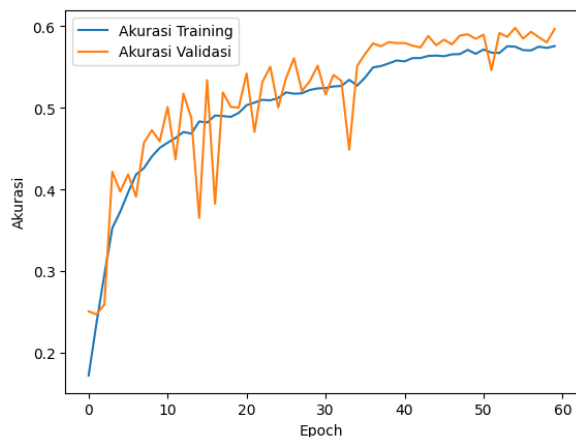
Tabel 3. Hasil Kinerja Model Menggunakan Convolutional Neural Network

| Batch Size | Epoch | Accuracy |
|------------|-------|----------|
| 64         | 30    | 56 %     |
| 64         | 60    | 60 %     |
| 128        | 30    | 55 %     |
| 128        | 60    | 60 %     |
| 256        | 30    | 53 %     |
| 256        | 57    | 58 %     |

Berdasarkan jumlah dataset pada Tabel 1, telah dilakukan enam kali percobaan pelatihan model dengan *batch size* dan *epoch* yang berbeda. Pada *batch size* 64 dengan *epoch* 30 dan *epoch* 60 menghasilkan akurasi sebesar 56% dan 60%. Pada *batch size* 128 dengan *epoch* 30 dan *epoch* 60 menghasilkan akurasi 55% dan 60%. Sedangkan *batch size* 256 dengan *epoch* 30 dan *epoch* 57 menghasilkan akurasi lebih rendah dari yang sebelumnya yaitu 53% dan 58%. Pada penelitian ini untuk membuat *training* model lebih efisien dan optimal maka pada proses pelatihan model dilakukan pengaturan *callbacks* dengan menggunakan *EarlyStopping*. Sehingga pada percobaan *batch size* 256 yang seharusnya melatih *epoch* sampai 60, namun karena tidak ada lagi peningkatan performa maka *epoch* berhenti pada *epoch* 57 dan menghasilkan akurasi 58%. Dari percobaan pelatihan model yang telah dilakukan, hasil akurasi tertinggi yaitu sebesar 60% yang dilakukan pada percobaan *batch size* 64 dan *batch size* 128 pada *epoch* 60. Hasil akurasi terendah yaitu pada percobaan *batch size* 256, *epoch* 30 sebesar 53%. Grafik yang dihasilkan oleh percobaan *batch size* 64 dan *batch size* 128 pada *epoch* 60 dapat dilihat dengan jelas pada Gambar 5 dan Gambar 6.

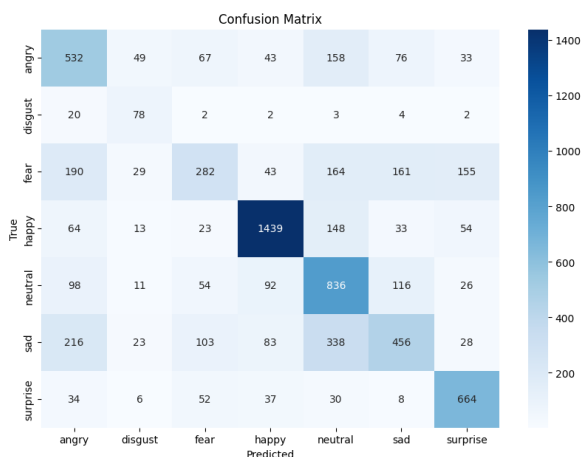


Gambar 5. Grafik Akurasi Batch Size 64 Epoch 60

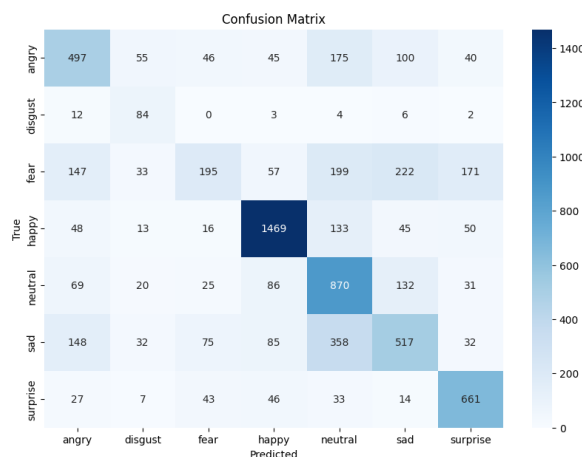


Gambar 6. Grafik Akurasi Batch Size 128 Epoch 60

Pada Gambar 5 menunjukkan bahwa pada awal *training*, epoch 0-20, akurasi validasi tampak fluktuatif dengan naik turun yang cukup tajam. Namun dengan bertambahnya *epoch*, setelah *epoch* ke 25 akurasi mulai stabil dan mendekati akurasi *training*. Hal ini menandakan bahwa model mulai melakukan generalisasi dengan lebih baik. Sedangkan pada Gambar 6, pada awal *training*, epoch 0-20 akurasi dan validasi sama-sama meningkat cukup cepat, meskipun akurasi validasi mengalami fluktuatif yang artinya masih dalam proses penyesuaian dalam data. Tetapi mulai dari *epoch* 20 grafik terlihat lebih stabil dan akurasi validasi terus mengalami peningkatan meskipun masih terdapat penurunan kecil. Pada *epoch* 40 keatas akurasi validasi cenderung lebih tinggi dari pada akurasi *training*. Hal ini menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap data yang baru. Berdasarkan dari kedua grafik yang dihasilkan grafik pada Gambar 6 lebih baik dari pada Gambar 5. Karena pada Gambar 6 stabilitas validasi lebih cepat tercapai, lebih sedikit fluktuatif dan model tidak *overfitting* serta model belajar lebih baik terhadap data yang baru. Hasil lain dari penelitian ini yaitu dari *confusion matrix* yang dapat dilihat pada Gambar 7 dan Gambar 8.



Gambar 7. Confusion Matrix Batch Size 64 Epoch 60



Gambar 8. Confusion Matrix Batch Size 128 Epoch 60

Hasil dari *confusion matrix* terlihat masih ada beberapa yang salah dalam mendeteksi. Pada Gambar 7 model, lebih seimbang dalam mendeteksi emosi. Sedangkan pada Gambar 8, model lebih kuat pada kelas yang lebih besar. Adapun hasil dari *classification report* dapat dilihat pada Gambar 9 dan Gambar 10.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| angry        | 0.46      | 0.56   | 0.50     | 958     |
| disgust      | 0.37      | 0.70   | 0.49     | 111     |
| fear         | 0.48      | 0.28   | 0.35     | 1024    |
| happy        | 0.83      | 0.81   | 0.82     | 1774    |
| neutral      | 0.50      | 0.68   | 0.57     | 1233    |
| sad          | 0.53      | 0.37   | 0.43     | 1247    |
| surprise     | 0.69      | 0.80   | 0.74     | 831     |
| accuracy     |           |        | 0.60     | 7178    |
| macro avg    | 0.55      | 0.60   | 0.56     | 7178    |
| weighted avg | 0.60      | 0.60   | 0.59     | 7178    |

Gambar 9. Classification Report Batch Size 64 Epoch 60

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| angry        | 0.52      | 0.52   | 0.52     | 958     |
| disgust      | 0.34      | 0.76   | 0.47     | 111     |
| fear         | 0.49      | 0.19   | 0.27     | 1024    |
| happy        | 0.82      | 0.83   | 0.82     | 1774    |
| neutral      | 0.49      | 0.71   | 0.58     | 1233    |
| sad          | 0.50      | 0.41   | 0.45     | 1247    |
| surprise     | 0.67      | 0.80   | 0.73     | 831     |
| accuracy     |           |        | 0.60     | 7178    |
| macro avg    | 0.55      | 0.60   | 0.55     | 7178    |
| weighted avg | 0.60      | 0.60   | 0.58     | 7178    |

Gambar 10. Classification Report Batch Size 128 Epoch 60

Berdasarkan hasil dari kedua *classification report* tersebut ada sedikit peningkatan di beberapa kelas tetapi ada juga beberapa kelas lain yang mengalami penurunan performa. Namun secara umum performa keseluruhan hampir sama, hanya saja ada sedikit perbedaan yang sangat kecil. Jika dibandingkan *classification report* pada Gambar 9 lebih baik dibandingkan dengan Gambar 10. Tetapi perbedaan yang muncul tidak terlalu signifikan.

#### 4. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan dengan mendeteksi ekspresi wajah manusia menggunakan metode *Convolutional Neural Network* dengan beberapa kali melakukan percobaan pelatihan model, hasil akurasi tertinggi yang dihasilkan yaitu sebesar 60% dengan percobaan pelatihan model *batch size* 64 dan *batch size* 128 dengan *epoch* 60. Namun jika dilihat berdasarkan dari grafik yang dihasilkan pelatihan model terbaik dilakukan dengan *batch size* 128 *epoch* 60. Hal tersebut dikarenakan stabilitas validasi lebih cepat tercapai, lebih sedikit fluktuatif dan model tidak *overfitting* serta model belajar lebih baik terhadap data yang baru. Jika dilihat dari *confusion matrix* yang dihasilkan *batch size* 64 *epoch* 60 model yang dihasilkan lebih seimbang dalam mendeteksi emosi wajah manusia. Sedangkan dari *classification report* yang dihasilkan tidak ada perbedaan yang signifikan. Secara umum performa yang dihasilkan hampir mirip. Namun *batch size* 64 *epoch* 60 sedikit lebih baik. Adapun saran untuk penelitian selanjutnya mungkin dapat dengan menambahkan *transfer learning* seperti *ResNet*, *MobileNetV2*, *EfficientNet* dll agar dapat mencapai akurasi yang lebih optimal dan menghasilkan performa yang lebih baik di semua kelas.

#### Daftar Rujukan

[1] D. Setiawan, S. Widodo, T. Ridwan, and R. Ambari, "Perancangan Deteksi Emosi Manusia berdasarkan Ekspresi Wajah Menggunakan Algoritma VGG16," *Syntax J. Inform.*, vol. 11, no. 01, pp. 01–11, 2022, doi: 10.35706/syji.v11i01.6594.

[2] N. Djamsi, D. R. Chandranegara, and Z. Sari, "Mendeteksi Ekspresi Wajah dengan Meninjau Iris Mata Menggunakan Metode Transformasi Hough dan K-Nearest," *REPOSITOR*, vol. 5, no. 1, pp. 575–580, 2023.

[3] Z. Pramuditha, Adeyuni, Suroso, and M. Fadhli, "Deteksi Wajah Dengan Model Arsitektur VGG 19 Pada Metode Convolutional Neural Network," *Sist. J. Sist. Inf.*, vol. 13, no. 5, pp. 1998–2007, 2024.

[4] A. S. Guntoro, E. Julianto, and D. Budiyo, "Pengenalan Ekspresi Wajah Menggunakan Convolutional Neural Network," *J. Inform. Atma Jogja*, vol. 3, no. 2, pp. 155–160, 2022, doi: 10.24002/jiaj.v3i2.6790.

[5] I. Azhar and Fitriyani, "Implementasi Algoritma Convolutional Neural Network Dalam Deteksi Emosi Manusia Berdasarkan Ekspresi Wajah," *eProsiding Tek. Inform.*, vol. 1, no. 1, pp. 112–118, 2020.

[6] M. A. Fadilla, H. Setiawan, and M. Ramadhan, "Implementasi Metode Convolutional Neural Network (CNN) Pada Sistem Deteksi Emosi Dari Ekspresi Wajah Manusia Dengan Aplikasi Android Sebagai Antarmuka Pengguna," *J. Ilmu Komput. dan Sist. Inf.*, vol. 9, no. 4, pp. 126–138, 2023.

[7] M. Y. A. Thoriq, K. E. Permana, and I. A. Siradjuddin, "Deteksi Wajah Manusia Berbasis One Stage Detector Menggunakan Metode You Only Look Once (Yolo)," *J. Teknoinfo*, vol. 17, no. 1, pp. 66–73, 2023, doi: 10.33365/jti.v17i1.1884.

[8] N. Dewi and F. Ismawan, "Implementasi Deep Learning Menggunakan Convolutional Neural Network Untuk Sistem Pengenalan Wajah," *Fakt. Exacta*, vol. 14, no. 1, pp. 34–43, 2021, doi: 10.30998/faktorexacta.v14i1.8989.

[9] N. R. Fauziyya and T. Rahajoeningroem, "Metoda Convolutional Neural Network (CNN) untuk Pendeteksi Tangga pada Alat Pemandu Arah bagi Penyandang Tunanetra," *Telekontran J. Ilm. Telekomun. Kendali dan*

*Elektron. Terap.*, vol. 8, no. 2, pp. 145–153, 2020, doi: 10.34010/telekontran.v8i2.4709.

[10] I. Maulana, N. Khairunisa, and R. Mufidah, "Deteksi Bentuk Wajah Menggunakan Convolutional Neural Network (CNN)," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 6, pp. 3348–3355, 2023, doi: 10.36040/jati.v7i6.8171.

[11] S. Zahrah, Azhar, and M. Abdi, "Sistem Deteksi Wajah Untuk Pencatatan Kehadiran Mahasiswa Di Kelas Menggunakan Metode Convolutional Neural Network," *JAISE J. Artif. Intell. Softw. Eng.*, vol. 2, no. 1, pp. 1–5, 2022.

[12] A. K. Muttaqin, H. Yuana, and M. T. Chulkamdi, "Implementasi Algoritma Convolutional Neural Network Untuk Pengenalan Ekspresi Wajah," *J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 8, no. 2, pp. 772–792, 2023, [Online]. Available: <https://tunasbangsa.ac.id/ejurnal/index.php/jurasik>.

[13] A. P. Saputri, A. Taqwa, and S. Soim, "Analisis Deteksi Objek Citra Digital Menggunakan Algoritma YOLO dan CNN dengan Arsitektur REPVGG Pada Sistem Pendeteksian dan Pengenalan Ekspresi Wajah," *Syntax Lit. J. Ilm. Indones.*, vol. 7, no. 9, pp. 13068–13080, 2022.

[14] J. Roosandriantini, R. Putranda, Y. Wahyuningsih, Y. Christella, Oktaviani, and E. Y. Christin, "Face Expression Recognizer Dengan Convolutiona Neural Network Untuk Membantu Penderita Autisme Mengenal Ekspresi Wajah Seseorang," *JITET (Jurnal Inform. dan Tek. Elektro Ter.)*, vol. 11, no. 3, pp. 336–342, 2023.

[15] P. Musa, W. K. Anam, S. B. Musa, W. Aryunani, R. Senjaya, and P. Sularsih, "Pembelajaran Mendalam Pengklasifikasi Ekspresi Wajah Manusia dengan Model Arsitektur Xception pada Metode Convolutional Neural Network," *Rekayasa. J. Sci. Technol.*, vol. 16, no. 1, pp. 65–73, 2023, doi: 10.21107/rekayasa.v16i1.16974.

[16] P. A. Nugroho, I. Fenriana, and R. Ariyanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia," *J. ALGOR*, vol. 2, no. 1, pp. 12–21, 2020.

[17] E. Y. Christin and Y. Wahyuningsih, "Implementasi Convolutional Neural Network Pada Pengenalan Ekspresi Wajah Untuk Anak Usia Dini," *Semin. Nas. Ilmu Terap.*, pp. 1–8, 2023.

[18] J. H. Niasmara and I. K. D. Nuryana, "Sistem Deteksi Ekspresi Siswa Dalam E-Learning Menggunakan Metode Convolutional Neural Network (CNN)," *JINACS (Journal Informatics Comput. Sci.)*, vol. 06, no. 02, pp. 551–556, 2024.

[19] F. Aure, R. Rokhana, R. Sigit, and B. S. B. Dewantara, "Fusi Algoritma K-Means dan CNN untuk Klasifikasi Emosi pada Anak," *Techo.COM*, vol. 22, no. 3, pp. 622–634, 2023, doi: 10.33633/tc.v22i3.8667.

[20] A. N. Ridho, G. Mellyka, F. Saputra, and A. P. A. Masa, "Implementasi Metode Convolutional Neural Network (CNN) Untuk Klasifikasi Gambar Mobil dan Motor Menggunakan Keras," *J. Jambo Digit.*, vol. 1, no. 1, pp. 54–64, 2024, doi: 10.1201/9781032622408-13.

[21] S. N. Amartama, A. N. Hidayah, P. K. Sari, and R. A. Ramadhani, "Implementasi Convolutional Neural Network (CNN) dalam Pengenalan Pola Penulisan Tangan," *Semin. Nas. Teknol. Sains*, vol. 3, no. 1, pp. 133–138, 2024, doi: 10.29407/stains.v3i1.4155.

[22] M. D. Darajat, Y. A. Sari, and R. C. Wihandika, "Convolutional Neural Network untuk Klasifikasi Citra Makanan Khas Indonesia," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 11, pp. 4764–4769, 2021, [Online]. Available: <http://j-ptiik.ub.ac.id>.

[23] A. S. Ritonga and I. Muhandhis, "Analisis dan Implementasi Metode Viola-Jones dan CNN pada Sistem Deteksi Kantuk Real-Time," *J. Ilm. IT CIDA Disem. Teknol. Inf.*, vol. 10, no. 1, pp. 53–66, 2024.

[24] A. L. Seandrio, A. H. Pratomo, and M. Y. Florestiyanto, "Implementation of Convolutional Neural Network ( CNN ) in Facial Expression Recognition Implementasi Convolutional Neural Network ( CNN ) Pada Pengenalan Ekspresi Wajah," *Telemat. J. Inform. dan Teknol. Inf.*, vol. 18, no. 2, pp. 211–221, 2021, doi:

- 
- [25] 10.31515/telematika.v18i2.4823.  
L. M. R. Rere, S. Usna, and Soegijanto, "Studi Pengenalan Ekspresi Wajah Berbasis Convolutional Neural Network," *Semin. Nas. Teknol. Inf. dan Komun. STI&K*, vol. 3, pp. 71–78, 2019.
- [26] M. Sambare, "FER-2013," *Kaggle*, 2020.  
<https://www.kaggle.com/datasets/msambare/fer2013>.