

Identifikasi Wajah Pengidap *Down Syndrome* Menggunakan Model CNN

Farnas Rozaan Iraquee (13520067)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13520067@std.stei.itb.ac.id

Abstrak— Sindrom Down, juga dikenal sebagai kondisi genetik trisomi, adalah gangguan genetik yang memengaruhi banyak orang. Kehadiran tambahan kromosom 21 menyebabkan penyakit genetik trisomi ini. Pengidap sindrom Down biasanya memiliki karakteristik fisik, mental, dan kognitif yang berbeda dengan orang normal. Karakteristik fisik yang berbeda tersebut, salah satunya terletak pada wajah mereka. Hal ini menyebabkan sindrom Down dapat didiagnosis dengan mendeteksi wajah, atau karakteristik wajah. Pada makalah ini, akan digunakan tiga buah model *Convolutional Neural Network* (CNN) untuk mengidentifikasi wajah orang pengidap sindrom Down tersebut. Pada tiap model tersebut kemudian akan dilakukan eksperimen dengan melakukan modifikasi terhadap beberapa parameter. Hasil yang didapatkan menunjukkan akurasi berkisar antara 70 hingga 80% dengan akurasi tertinggi sebesar 82,61%. Rata-rata akurasi tertinggi diperoleh oleh model kedua dengan rata-rata sebesar 76,53%. Konfigurasi parameter terbaik diperoleh dengan membagi data sebesar 81/19%, menggunakan mode warna RGB, dan melakukan augmentasi terhadap data latih.

Kata kunci—Sindrom Down; karakteristik wajah; CNN

I. PENDAHULUAN

Saat ini, teknologi informasi sedang mengalami peningkatan yang cukup pesat. Kehadirannya telah memberikan banyak kontribusi bagi manusia. Dalam bidang kecerdasan buatan, salah satu perkembangannya adalah kemampuan sistem untuk mengidentifikasi wajah manusia dengan tingkat akurasi yang cukup tinggi. Identifikasi wajah ini banyak memberikan manfaat bagi kehidupan manusia, seperti dalam bidang keamanan, membantu pengawasan ataupun membantu dalam pengembangan teknologi kesehatan.

Sindrom Down, juga dikenal sebagai kondisi genetik trisomi, adalah gangguan genetik yang memengaruhi banyak orang. Kehadiran tambahan kromosom 21 menyebabkan penyakit genetik trisomi ini. Kromosom tambahan tersebut meningkatkan jumlah protein tertentu, mengganggu pertumbuhan alami tubuh. Hal ini juga dapat mengakibatkan perubahan yang sudah ditentukan dalam perkembangan otak. Kelainan ini juga dapat menyebabkan keterlambatan perkembangan, hambatan belajar, masalah jantung, dan kanker darah. Ras, negara, agama, atau tingkat sosial ekonomi tidak memengaruhi penyakit ini.

Selain itu, pengidap sindrom Down memiliki ciri-ciri fisik seperti bentuk kepala yang lebih kecil dibandingkan orang

normal, area datar di tengkuk, ubun-ubun yang lebih besar, bentuk mata sipit, dan lipatan tengah yang membentuk mulut kecil dengan lidah yang panjang, sehingga membuatnya tampak menonjol. Ciri-ciri fisik lainnya dari penderita sindrom Down antara lain: ciri-ciri wajah datar, kepala dan telinga kecil, garis leher pendek, lidah besar, mata miring ke atas, tonus otot buruk, dan badan pendek. Selain itu, ciri-ciri mental sindrom Down meliputi rentang perhatian yang pendek, perilaku impulsif, pembelajaran yang lamban, dan perkembangan bahasa dan bicara yang tertunda.



Gambar I.1 Anak pengidap sindrom Down

(sumber: <https://www.halodoc.com/kesehatan/sindrom-down>)

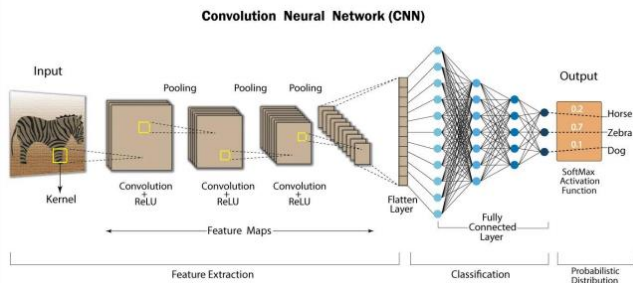
Masalah kognitif, seperti keterlambatan intelektual dan perkembangan, ketidakmampuan belajar, dan gangguan bicara, merupakan hal yang biasa terjadi pada sindrom Down. Sindrom Down merusak hipokampus, yang penting untuk memori dan pembelajaran. Orang yang mengidap sindrom Down lebih mungkin mengalami beberapa masalah kesehatan, seperti penyakit tiroid, leukemia, obesitas, sembelit kronis, apnea tidur, penglihatan buruk, katarak, strabismus, anemia, cacat jantung bawaan, dan gangguan pendengaran. Sindrom Down dapat didiagnosis dengan mendeteksi wajah, atau karakteristik wajah.

Pada makalah ini, akan dibahas penerapan model *Convolutional Neural Network* (CNN) dalam identifikasi wajah orang-orang yang mengidap sindrom Down. CNN merupakan salah satu jenis arsitektur jaringan saraf tiruan yang telah terbukti efektif dalam tugas pengenalan pola pada gambar, khususnya dalam domain pengolahan citra. Dengan memanfaatkan keunggulan teknologi ini, akan dibangun sistem yang mampu mengenali ciri-ciri khas wajah orang-orang dengan sindrom Down.

II. LANDASAN TEORI

A. Convolution Neural Network (CNN)

Convolutional Neural Network (CNN atau ConvNet) adalah algoritma pembelajaran mendalam yang populer saat ini. CNN dapat disebut juga jaringan syaraf tiruan yang melibatkan konvolusi (CNN = ANN + konvolusi). CNN ini umumnya digunakan untuk memproses data yang memiliki topologi, seperti grid. Contoh data berbentuk grid adalah citra. CNN merupakan arsitektur jaringan untuk pembelajaran mendalam yang belajar langsung dari data, dengan menghilangkan kebutuhan untuk melakukan ekstraksi fitur secara manual.



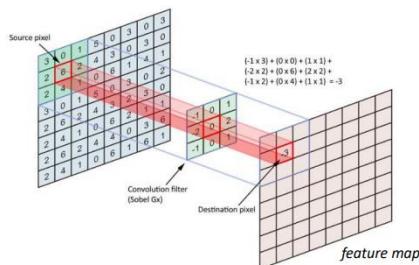
Gambar II.1 Ilustrasi CNN

(sumber: <https://developersbreach.com/convolution-neural-network-deep-learning/>)

CNN itu sendiri secara umum terdiri dari tiga lapisan utama, yaitu sebagai berikut:

1. Convolutional layer

Layer ini adalah inti dari CNN. Konvolusi digunakan untuk mengekstraksi fitur-fitur penting dari gambar dengan memindai *filter* atau kernel ke seluruh citra. Proses konvolusi memungkinkan jaringan untuk mendeteksi pola-pola seperti tepi, sudut, atau tekstur pada tingkat yang berbeda. Tiap *filter* tersebut menghasilkan luaran yang disebut *feature map*. Terdapat beberapa *hyperparameter* yang mempengaruhi ukuran *feature map*, yaitu jumlah *filter*, *stride*, dan *padding*.

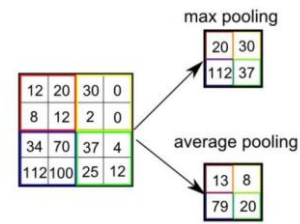


Gambar II.2 Ilustrasi convolutional layer

(sumber: Bahan Kuliah IF4073 Interpretasi dan Pengolahan Citra)

2. Pooling layer

Layer ini digunakan untuk mengurangi dimensi spasial gambar dan jumlah parameter dalam jaringan. *Pooling* umumnya dilakukan dengan operasi seperti maksimum (*max pooling*) atau rata-rata (*average pooling*) di wilayah tertentu.

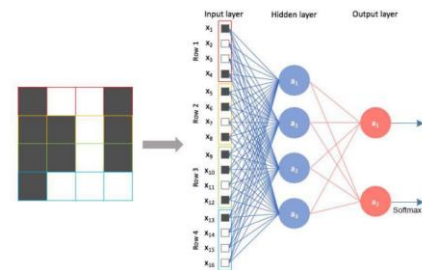


Gambar II.3 Ilustrasi pooling layer

(sumber: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>)

3. Fully-connected layer

Lapisan terakhir di dalam CNN adalah *fully-connected layer* (FC) yang menghasilkan vektor dimensi K, dalam hal ini K adalah jumlah kelas yang dapat diprediksi oleh jaringan. Vektor ini berisi probabilitas untuk setiap kelas dari setiap gambar yang diklasifikasikan. Lapisan terakhir dari arsitektur CNN menggunakan fungsi *softmax* untuk menyediakan luaran klasifikasi.



Gambar II.4 Ilustrasi FC layer

(sumber: Bahan Kuliah IF4073 Interpretasi dan Pengolahan Citra)

Selain itu, terdapat beberapa *layer* tambahan lainnya yang bertujuan untuk mempercepat pelatihan, melakukan regularisasi, atau normalisasi. *Layer-layer* tersebut adalah sebagai berikut:

1. ReLU layer

ReLU adalah *layer* tambahan yang memungkinkan pelatihan yang lebih cepat dan efektif dengan memetakan nilai negatif ke nol dan mempertahankan nilai positif. Pada dasarnya ReLU adalah operasi per-pixel dengan cara mengganti nilai negatif pixel di dalam *feature map* menjadi nol.

2. Dropout layer

Dropout adalah teknik regularisasi yang secara acak menghilangkan sebagian dari unit (neuron) dalam

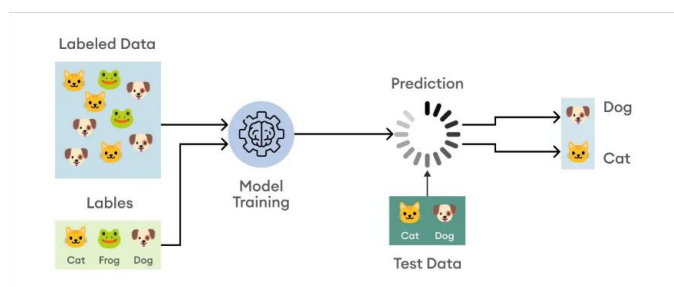
jaringan pada setiap iterasi pelatihan. Ini membantu mencegah *overfitting* dan meningkatkan generalisasi model.

3. Batch Normalization layer (BatchNorm)

BatchNorm adalah teknik normalisasi yang diterapkan pada jaringan saraf tiruan untuk meningkatkan stabilitas dan kecepatan konvergensi selama pelatihan. BatchNorm biasanya digunakan pada *layer-layer* dalam sebuah jaringan, seperti *fully-connected layer* atau *layer* konvolusi pada CNN.

B. Image Classification

Image classification adalah tugas dalam bidang pengolahan citra dan kecerdasan buatan yang bertujuan untuk mengklasifikasikan objek atau konten dalam suatu gambar ke dalam kategori atau kelas tertentu. Tujuan utama dari *image classification* adalah membuat model yang mampu mengidentifikasi dan membedakan objek atau pola visual di dalam gambar secara akurat.



Gambar II.5 Ilustrasi *image classification*

(sumber: <https://www.superannotate.com/blog/image-classification-basics>)

Secara umum, berikut adalah beberapa langkah dan konsep utama dalam proses *image classification*:

1. Dataset

Image classification memerlukan dataset yang mencakup berbagai gambar yang mewakili kelas-kelas yang ingin diidentifikasi. Setiap gambar dalam dataset biasanya dilabeli dengan kelas yang sesuai. Dataset kemudian dibagi menjadi set pelatihan (*training set*) dan set pengujian (*test set*). Set pelatihan digunakan untuk melatih model, sementara set pengujian digunakan untuk mengukur performa model pada data yang tidak pernah dilihat sebelumnya.

2. Pre-processing

Pre-processing dapat dilakukan dengan melakukan beberapa aksi, seperti normalisasi, *resizing*, dan augmentasi data.

3. Model arsitektur

Arsitektur CNN umumnya digunakan untuk tugas *image classification* karena kemampuannya dalam mengekstraksi fitur hierarkis dari gambar. CNN terdiri dari lapisan konvolusi, aktivasi ReLU, *pooling*, dan lapisan *fully-connected*. Selain itu, model lainnya yang

biasa digunakan untuk *image classification*, antara lain adalah *logistic regression*, KNN, SVM, dan *decision tree*.

4. Pelatihan model

Pelatihan model dilakukan dengan menetapkan beberapa hal, seperti *loss function* ataupun algoritma optimasi. *Loss function* digunakan untuk mengukur seberapa baik model memprediksi kelas dari gambar dibandingkan dengan label yang sebenarnya. Algoritma optimasi, seperti *Stochastic Gradient Descent* (SGD) atau Adam, digunakan untuk meminimalkan *loss function* dengan memperbarui bobot dan bias model.

5. Validasi dan tuning

Setelah pelatihan, model diuji pada set validasi untuk memastikan bahwa tidak terjadi *overfitting* dan untuk menyesuaikan *hyperparameter* jika diperlukan. Pengaturan parameter, seperti *learning rate*, jumlah lapisan, dan ukuran batch, akan di-*tuning* untuk meningkatkan performa model.

6. Evaluasi model

Metrik seperti akurasi, presisi, *recall*, F1-score, dan *confusion matrix* dapat digunakan untuk mengevaluasi sejauh mana model dapat mengklasifikasikan gambar dengan benar pada set pengujian.

7. Inferensi

Setelah melalui proses pelatihan dan evaluasi, model dapat digunakan untuk memprediksi kelas dari gambar baru yang belum pernah dilihat sebelumnya.

Image classification dapat digunakan dalam berbagai aplikasi, termasuk pengenalan wajah, deteksi objek, pemantauan keamanan, dan banyak lagi. Kemajuan terbaru dalam *deep learning*, terutama dengan menggunakan arsitektur seperti CNN, telah meningkatkan tingkat akurasi *image classification*.

III. IMPLEMENTASI

Dalam proses pengimplementasiannya terdapat beberapa langkah yang dilakukan. Berikut adalah langkah-langkah yang dilakukan tersebut:

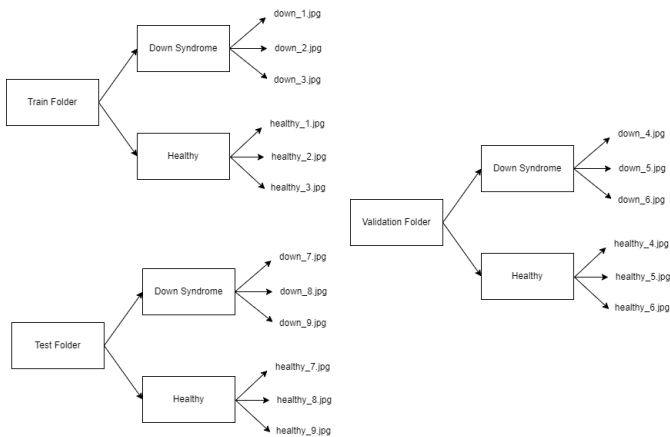


Gambar III.1 Langkah implementasi

A. Akuisisi Dataset

Dataset yang digunakan untuk proses pelatihan, validasi, sekaligus *testing* diambil dari *platform* Kaggle sebagai berikut: [dataset](#). Dataset tersebut terdiri dari total 3.000 gambar, dengan rincian 1.500 gambar berlabel orang sindrom Down dan 1.500 sisanya berlabel orang sehat. Gambar-gambar di dalam dataset tersebut memiliki variasi yang cukup besar dalam hal pose, pencahayaan, latar belakang, dll.

Selanjutnya akan dilakukan restrukturisasi folder dataset tersebut. Restrukturisasi ini dilakukan untuk mempermudah proses pelatihan serta evaluasi model nantinya. Restrukturisasi dilakukan dengan membagi dataset ke dalam tiga set, yaitu *training*, *validation*, dan *test set*. Struktur folder yang baru setelah proses restrukturisasi folder tersebut kurang lebih adalah sebagai berikut:



Gambar III.2 Struktur Folder Data

B. Pre-processing

Tahap *pre-processing* yang akan dilakukan pada data, meliputi normalisasi, *resizing*, dan augmentasi data. Proses normalisasi data tersebut dilakukan dengan melakukan *rescaling* pada gambar sehingga memiliki skala yang seragam, yaitu dalam rentang 0 hingga 1. Kemudian, tiap gambar akan di-*resize* dengan ukuran yang sama, yaitu 256×256 .

Selanjutnya, akan dilakukan proses augmentasi pada data latih berupa operasi *shear* dan *zoom* secara acak serta pembalikan horizontal. Proses augmentasi ini dapat secara bebas dilakukan dengan parameter yang bervariasi. Tujuan dari proses augmentasi ini adalah untuk meningkatkan variasi / keberagaman dari data latih sehingga model dapat menggeneralisasi data yang tidak terlihat dengan lebih baik. Tahap *pre-processing* ini memanfaatkan kelas *ImageDataGenerator* pada TensorFlow. Dengan menggunakan kelas tersebut, gambar akan lebih mudah diproses secara dinamis. Berikut adalah contoh *pre-processing* data:

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
)
valid_datagen = ImageDataGenerator(rescale=1./255)
```

Gambar III.3 Contoh *Pre-processing* data

C. Pelatihan Model

Terdapat tiga arsitektur model CNN yang akan diimplementasikan pada tugas kali ini. Masing-masing memiliki konfigurasi *layer* yang berbeda-beda. Hal ini bertujuan untuk mencari kinerja model yang paling optimal.

Berikut adalah masing-masing arsitektur dari ketiga model tersebut:

1. Model CNN 1

Model ini terdiri dari 10 *layer*, dengan tiga *layer* konvolusi, tiga *layer* *max pooling*, satu *layer* *flatten*, satu FC (*dense*) *layer* dengan *dropout layer*, dan satu *dense layer* dengan fungsi aktivasi *softmax* sebagai *layer* klasifikasi. Berikut adalah rinciannya:

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_1 (Conv2D)	(None, 125, 125, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_2 (Conv2D)	(None, 60, 60, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 128)	0
flatten (Flatten)	(None, 115200)	0
dense (Dense)	(None, 128)	14745728
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

Total params: 14839234 (56.61 MB)
 Trainable params: 14839234 (56.61 MB)
 Non-trainable params: 0 (0.00 Byte)

Gambar III.4 Arsitektur Model CNN 1

2. Model CNN 2

Model ini terdiri dari 14 *layer*, dengan tiga *layer* konvolusi dengan *BatchNorm layer*, tiga *layer* *max pooling*, satu *layer* *flatten*, satu FC (*dense*) *layer* dengan *BatchNorm* dan *dropout layer*, dan satu *dense layer* dengan fungsi aktivasi *softmax* sebagai *layer* klasifikasi. Berikut adalah rinciannya:

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 254, 254, 32)	896
batch_normalization (Batch Normalization)	(None, 254, 254, 32)	128
max_pooling2d_3 (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_4 (Conv2D)	(None, 125, 125, 64)	18496
batch_normalization_1 (Batch Normalization)	(None, 125, 125, 64)	256
max_pooling2d_4 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_5 (Conv2D)	(None, 60, 60, 128)	73856
batch_normalization_2 (Batch Normalization)	(None, 60, 60, 128)	512
max_pooling2d_5 (MaxPooling2D)	(None, 30, 30, 128)	0
flatten_1 (Flatten)	(None, 115200)	0
dense_2 (Dense)	(None, 128)	14745728
batch_normalization_3 (Batch Normalization)	(None, 128)	512
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 2)	258

Total params: 14840642 (56.61 MB)
 Trainable params: 14839938 (56.61 MB)
 Non-trainable params: 704 (2.75 KB)

Gambar III.5 Arsitektur Model CNN 2

3. Model CNN 3

Model ini terdiri dari 10 *layer*, dengan dua *layer* konvolusi dengan BatchNorm *layer*, dua *layer* *max pooling*, satu *layer* *flatten*, satu FC (*dense*) *layer* dengan BatchNorm *layer*, dan satu *dense* *layer* dengan fungsi aktivasi *softmax* sebagai *layer* klasifikasi. Berikut adalah rinciannya:

```

Model: "sequential_2"
Layer (type)                Output Shape              Param #
-----
conv2d_6 (Conv2D)           (None, 254, 254, 32)     896
batch_normalization_4 (Bat (None, 254, 254, 32)     128
chNormalization)
max_pooling2d_6 (MaxPoolin (None, 127, 127, 32)     0
g2D)
conv2d_7 (Conv2D)           (None, 125, 125, 64)    18496
batch_normalization_5 (Bat (None, 125, 125, 64)    256
chNormalization)
max_pooling2d_7 (MaxPoolin (None, 62, 62, 64)     0
g2D)
flatten_2 (Flatten)         (None, 246016)           0
dense_4 (Dense)             (None, 128)              31498176
batch_normalization_6 (Bat (None, 128)              512
chNormalization)
dense_5 (Dense)             (None, 2)                258
-----
Total params: 31510722 (120.20 MB)
Trainable params: 31510274 (120.20 MB)
Non-trainable params: 448 (1.75 KB)
    
```

Gambar III.6 Arsitektur Model CNN 3

Setelah model dibangun, selanjutnya akan dilakukan pelatihan terhadap model tersebut. Pada tugas kali ini, akan digunakan algoritma optimisasi berupa Adam, *loss function* berupa *categorical crossentropy*, dan metrik evaluasi berupa akurasi. Proses pelatihan akan dilakukan sebanyak 20 *epoch* dengan ukuran *batch* sebesar 32. Dengan langkah tiap *epoch* sebanyak total data latih / validasi dibagi dengan ukuran *batch*.

D. Evaluasi Model

Setelah dilakukan pelatihan terhadap model, akan dilakukan evaluasi terhadap model dengan menggunakan *training set*. Metrik evaluasi yang akan digunakan, seperti yang sudah disebutkan sebelumnya, adalah akurasi.

E. Inferensi

Setelah melalui proses pelatihan dan evaluasi, model dapat digunakan untuk memprediksi kelas dari gambar baru yang belum pernah dilihat sebelumnya. Prediksi dapat dilakukan menggunakan gambar *batch* dari *test generator* / *generator* baru atau bisa juga dengan meng-*upload* gambar baru ke Google Colab.

Proses implementasi model ini dijalankan pada Google Colab dengan memanfaatkan *library* TensorFlow. Hasil implementasinya dapat dilihat pada pranala berikut: [Down Syndrome CNN](#).

IV. HASIL EKSPERIMEN

Pada tugas kali ini, dilakukan eksperimen pada model dengan melakukan beberapa modifikasi. Modifikasi tersebut dilakukan pada saat proses *splitting* data dan *pre-processing*

data. Berikut adalah rincian pengaturan parameter eksperimen yang dilakukan:

Tabel IV.1 Pengaturan Parameter Eksperimen

Parameter	Pengaturan
Rasio <i>Splitting</i> Data	<i>Test</i> : 10% dari data awal <i>Validation</i> : 10% dari data setelah proses pertama <i>Training</i> : <i>Validation/Test</i> (81 : 19)
	<i>Test</i> : 20% dari data awal <i>Validation</i> : 20% dari data setelah proses pertama <i>Training</i> : <i>Validation/Test</i> (64 : 36)
Mode Warna	Grayscale
	RGB
Augmentasi Data Latih	Iya
	Tidak

Eksperimen akan dilakukan dengan melakukan pelatihan dan evaluasi terhadap model pada setiap konfigurasi dari kombinasi pengaturan parameter eksperimen di atas. Jadi, total ada delapan konfigurasi eksperimen yang mungkin untuk setiap model. Metrik evaluasi yang menjadi bahan perbandingan adalah akurasi.

Berikut adalah perbandingan hasil evaluasi untuk setiap konfigurasi beserta rata-rata untuk tiap model dan tiap pengaturan parameter:

Tabel IV.2 Perbandingan Hasil Evaluasi dan Rata-rata

Parameter			Akurasi (%)		
Ratio training & valid/test	Warna	Augmentasi	Model 1	Model 2	Model 3
81/19	Gray	Ya	75.92	76.59	79.6
		Tidak	74.25	75.25	76.25
	RGB	Ya	82.61	81.61	72.58
		Tidak	74.92	70.9	80.6
64/36	Gray	Ya	74.96	73.29	73.62
		Tidak	73.79	77.8	77.8
	RGB	Ya	78.13	80.47	72.12
		Tidak	77.13	76.29	78.96
Rata-rata total			76.46375	76.525	76.44125
Rata-rata 81/19			76.75666667		
Rata-rata 64/36			76.19666667		
Rata-rata Gray			75.76		
Rata-rata RGB			77.19333333		
Rata-rata Augmentasi			76.79166667		
Rata-rata Tidak Augmentasi			76.16166667		

Berdasarkan hasil di atas, dapat dilihat bahwa nilai akurasi terbesar diperoleh oleh model 1 dengan konfigurasi parameter rasio *splitting* data 81/19, mode warna RGB, dan dilakukan augmentasi data latih, yaitu 82,61%. Namun, ternyata rata-rata akurasi tertinggi diperoleh oleh model 2 (76,53%) dengan selisih rata-rata dengan model lain yang tidak jauh berbeda, yaitu hanya berbeda satu angka di belakang koma saja.

Kemudian, untuk parameter rasio *splitting* data, ternyata *splitting* data sebesar 81/19 memiliki rata-rata akurasi yang lebih tinggi daripada rasio *splitting* data 64/36, yaitu sebesar 76,76%. Hal ini dikarenakan data latih yang digunakan lebih banyak sehingga memungkinkan model untuk belajar dengan lebih baik sehingga kemampuan generalisasinya akan meningkat.





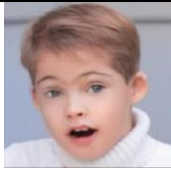
Selanjutnya, untuk parameter mode warna, konfigurasi dengan mode warna RGB menghasilkan rata-rata akurasi yang lebih baik daripada konfigurasi dengan mode warna *grayscale*, yaitu 77,19%. Hal ini bisa terjadi karena dengan menggunakan mode warna RGB model dapat belajar hingga ke dalam tiga kanal warna. Selain itu, dengan menggunakan mode warna variasi gambar tentu akan lebih besar, mengingat warna yang terdapat dalam tiap gambar pasti berbeda-beda. Dengan begitu, model akan dapat menggeneralisasi data yang tidak terlihat dengan lebih baik.

Berikutnya, untuk parameter dilakukan atau tidaknya augmentasi pada data latih, rata-rata akurasi pada konfigurasi yang melakukan augmentasi ternyata lebih baik, yaitu sebesar 76,79%. Seperti yang sudah dijelaskan sebelumnya, augmentasi data bertujuan untuk meningkatkan variasi data agar model dapat menggeneralisasi data yang tidak terlihat dengan lebih baik dan mencegah *overfitting*. Apabila tidak dilakukan augmentasi pada data, terlebih lagi pada eksperimen kali ini pelatihan model dilakukan sebanyak 20 *epoch*, maka model cenderung *overfit*, yaitu memiliki akurasi *training* yang hampir sempurna, tetapi kemampuan generalisasinya cukup buruk.


Terlihat pada hasil eksperimen bahwa nilai akurasi yang dihasilkan berkisar antara 70 hingga 80%. Seperti yang sudah disebutkan pada bagian sebelumnya bahwa gambar-gambar yang ada di dalam dataset ini memiliki variasi yang cukup besar dalam hal pose, pencahayaan, latar belakang, dll. Jadi, meskipun terkadang nilai akurasi *training*-nya tinggi, tetapi model masih belum bisa dengan baik mengidentifikasi data yang tidak terlihat, atau bisa dibilang *overfit*. Oleh karena itu, perlu dilakukan pemrosesan pada gambar secara lebih lanjut agar model dapat memiliki kinerja yang lebih baik.

Setelah melakukan evaluasi model, akan dilakukan juga proses inferensi / prediksi untuk menguji kemampuan model. Berikut akan dilakukan contoh proses inferensi dengan masukan dari *batch* gambar yang dihasilkan *generator* dan gambar dari hasil *upload* ke Google Colab:

1. *Batch* gambar dari *generator* (dibatasi hingga 5 gambar saja)

Gambar	Prediksi Kelas	Kelas Asli
	<i>Healthy</i>	<i>Healthy</i>
	<i>Healthy</i>	<i>Healthy</i>
	<i>Down Syndrome</i>	<i>Down Syndrome</i>
	<i>Down Syndrome</i>	<i>Healthy</i>
	<i>Down Syndrome</i>	<i>Down Syndrome</i>

2. Gambar yang di-*upload* ke Google Colab

Gambar	Prediksi Kelas	Kelas Asli
	<i>Down Syndrome</i>	<i>Down Syndrome</i>

Dari hasil inferensi terlihat di atas bahwa model sudah cukup baik dalam mengidentifikasi apakah seseorang mengidap sindrom Down atau tidak. Meskipun ada satu gambar yang prediksinya masih salah.

V. KESIMPULAN

Pada tugas kali ini, dibangun 3 buah model CNN untuk dapat menjalankan tugas mengidentifikasi wajah pengidap sindrom Down. Kemudian, akan dilakukan eksperimen pada tiap model tersebut dengan melakukan modifikasi parameter. Hasil yang didapatkan menunjukkan akurasi berkisar antara 70 hingga 80% dengan akurasi tertinggi sebesar 82,61%. Rata-rata akurasi tertinggi diperoleh oleh model kedua dengan rata-rata sebesar 76,53%. Konfigurasi parameter terbaik diperoleh dengan membagi data sebesar 81/19%, menggunakan mode warna RGB, dan melakukan augmentasi terhadap data latih. Variasi gambar yang cukup besar dalam hal pose, pencahayaan, latar belakang, dll. menyebabkan hasilnya belum sepenuhnya optimal. Oleh karena itu, perlu dilakukan pemrosesan pada gambar secara lebih lanjut agar model tidak mengalami *overfitting* serta dapat memiliki kinerja yang lebih baik.

UCAPAN TERIMA KASIH

Penulis ingin menyampaikan rasa syukur kepada Allah SWT karena berkat rahmat dan karunia-Nya, penulis berhasil menyelesaikan makalah berjudul "Identifikasi Wajah Pengidap Down Syndrome Menggunakan Model CNN" ini dengan tepat waktu. Penulis juga ingin mengungkapkan terima kasih kepada orang tua yang selalu memberikan dukungan selama penulis menempuh masa perkuliahan di ITB. Selain itu, penulis mengucapkan terima kasih kepada Bapak Rinaldi Munir atas bimbingan yang diberikan selama satu semester dalam

mengikuti mata kuliah IF4073 Interpretasi dan Pengolahan Citra.

REFERENSI

- [1] Munir, Rinaldi. (2023). Bahan Kuliah IF4073 Interpretasi dan Pengolahan Citra. Program Studi Informatika ITB.
- [2] P. N. Alexandrov, M. E. Percy, and W. J. Lukiw, "Chromosome 21-Encoded microRNAs (mRNAs): Impact on Down's Syndrome and Trisomy-21 Linked Disease," *Cell. Mol. Neurobiol.*, vol. 38, no. 3, pp. 769–774, Jul. 2017.
- [3] V. Dima, A. Ignat, and C. Rusu, "Identifying Down Syndrome Cases by Combined Use of Face Recognition Methods," in *Advances in Intelligent Systems and Computing*, 2018, pp. 472–482.
- [4] <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html> diakses pada 18 Desember 2023

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Desember 2023



Farnas Rozaan Iraquee (13520067)