

# Utilization of ResNet Architecture and Transfer Learning Method in the Classification of Faces of Individuals with Down Syndrome

Made Doddy Adi Pranatha<sup>1\*</sup>, Gede Herdian Setiawan<sup>2\*\*</sup>, M Azman Maricar<sup>3\*\*</sup>

\*Program Studi Bisnis Digital, Fakultas Bisnis dan Vokasi, Institut Teknologi dan Bisnis STIKOM Bali

\*\*Program Studi Sistem Komputer, Fakultas Informatika dan Komputer, Institut Teknologi dan Bisnis STIKOM Bali

[doddy@stikom-bali.ac.id](mailto:doddy@stikom-bali.ac.id)<sup>1</sup>, [herdian@stikom-bali.ac.id](mailto:herdian@stikom-bali.ac.id)<sup>2</sup>, [azman@stikom-bali.ac.id](mailto:azman@stikom-bali.ac.id)<sup>3</sup>

## Article Info

### Article history:

Received 2024-09-11

Revised 2024-09-29

Accepted 2024-10-01

### Keyword:

CNN,  
Deep Learning,  
Down Syndrome,  
Image Classification,  
ResNet.

## ABSTRACT

Classifying the faces of individuals with Down Syndrome poses a significant challenge in image processing and genetic anomaly detection. This study leverages the ResNet34 architecture and transfer learning methods to improve classification accuracy for Down Syndrome facial recognition. Three experiments were conducted, varying the batch size, learning rate, and number of epochs. In the first experiment, the model achieved an accuracy of 82.83%, precision of 0.8362, recall of 0.8350, and an F1 score of 0.8348, showing promising performance but falling short of the target accuracy of 85%. The second experiment yielded the best results, with an accuracy of 87.88%, precision of 0.8956, recall of 0.8956, and an F1 score of 0.8956, indicating an optimal balance between correct predictions and errors. The third experiment resulted in the lowest accuracy, at 80.47%, with a precision of 0.8272, recall of 0.8249, and an F1 score of 0.8247, signifying a decline in performance compared to the other trials. Among the three experiments, the best configuration was achieved in the second trial, as the high recall value is crucial in medical contexts to ensure that as many individuals with Down Syndrome are correctly detected as possible, minimizing the risk of serious consequences due to false negatives.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

## I. PENDAHULUAN

Dalam beberapa tahun terakhir, kemajuan dalam bidang deep learning, khususnya *Convolutional Neural Network* (CNN), telah membuka peluang besar dalam aplikasi pengolahan citra medis, termasuk dalam klasifikasi wajah penderita *Down Syndrome*. *Down Syndrome* adalah kondisi genetik yang disebabkan oleh kelebihan kromosom 21 yang menyebabkan perubahan fisik dan kognitif pada penderitanya. Salah satu karakteristik fisik yang menonjol dari penderita *Down Syndrome* adalah perubahan pada morfologi wajah, yang dapat digunakan sebagai tanda untuk melakukan deteksi dini.

Penelitian ini bertujuan untuk mengeksplorasi pemanfaatan arsitektur ResNet dan metode *transfer learning* dalam mengklasifikasikan wajah penderita *Down Syndrome*. He et al. [1] memperkenalkan arsitektur ResNet, yang secara signifikan mengurangi masalah *vanishing gradient* pada

jaringan saraf tiruan dalam. Arsitektur ini memungkinkan pembuatan jaringan yang sangat dalam dengan menambahkan *shortcut connections*. ResNet telah menjadi model standar untuk berbagai tugas klasifikasi gambar dan telah diaplikasikan dalam pengenalan wajah.

Berbagai penelitian sebelumnya telah menyoroti efektivitas *transfer learning* dalam tugas-tugas klasifikasi gambar dengan *dataset* terbatas. Yosinski et al. [2] menunjukkan bahwa fitur yang dipelajari oleh model *deep learning* dari *dataset* besar, seperti ImageNet, dapat ditransfer untuk menyelesaikan masalah klasifikasi dengan *dataset* yang lebih kecil. Dalam konteks klasifikasi wajah *Down Syndrome*, pendekatan *transfer learning* memungkinkan model memanfaatkan pengetahuan umum yang diperoleh dari dataset umum seperti ImageNet, sehingga meningkatkan performa ketika diaplikasikan pada *dataset* spesifik dengan jumlah data terbatas.

Huang et al. [3] memperkenalkan DenseNet, yang menekankan konektivitas yang lebih rapat antara layer-layer dalam jaringan. Meskipun arsitektur ini efektif dalam beberapa tugas klasifikasi, Zhu et al. [4] dalam penelitiannya menunjukkan bahwa ResNet tetap lebih efisien secara komputasional dan lebih unggul dalam tugas klasifikasi gambar medis.

Penelitian terkait lainnya oleh Voulodimos et al. [5] menunjukkan bahwa penggunaan CNN dalam tugas klasifikasi gambar medis telah memberikan hasil yang sangat baik, dengan CNN mampu mengenali pola-pola visual yang kompleks. Pan et al. [6] menambahkan bahwa penggunaan transfer learning dalam tugas-tugas pengenalan citra medis memberikan peningkatan akurasi yang signifikan.

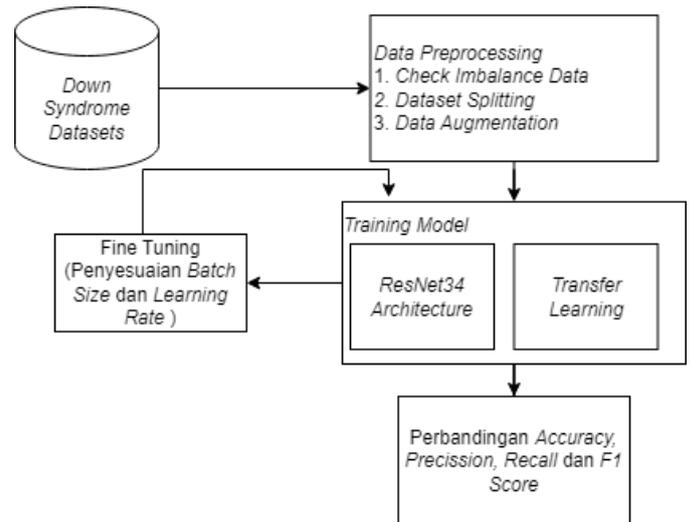
Dalam penelitian ini, kami melakukan tiga percobaan dengan konfigurasi *hyperparameter* yang berbeda untuk mengeksplorasi performa ResNet dalam klasifikasi wajah penderita *Down Syndrome*. Kami menguji variasi *batch size*, *learning rate*, dan *num epochs* untuk menemukan konfigurasi yang optimal. Percobaan ini juga didasarkan pada penelitian sebelumnya yang menunjukkan bahwa *tuning hyperparameter* secara tepat dapat secara signifikan mempengaruhi performa model *deep learning* [7].

Selain itu, penelitian sebelumnya yang relevan telah menunjukkan bahwa model CNN dapat digunakan untuk klasifikasi wajah penderita *Down Syndrome* dengan hasil yang cukup baik. Rozaan [8] menggunakan CNN untuk mendeteksi *Down Syndrome* dengan hasil akurasi hingga 82,61% dalam salah satu percobaannya. Dengan memanfaatkan *pre-processing* dan augmentasi data, penelitian tersebut memberikan hasil yang menjanjikan dalam mendeteksi kelainan genetik melalui analisis wajah.

Penelitian ini menggunakan *dataset* terbaru dari Kaggle[9] dimana *dataset* tersebut adalah *dataset down syndrome* terbaru yang terdiri dari 3000 data dan diharapkan dapat memberikan kontribusi pada pengembangan sistem deteksi otomatis untuk penderita *Down Syndrome*, yang dapat digunakan sebagai alat bantu diagnosis awal yang lebih cepat dan akurat.

**II. METODE PENELITIAN**

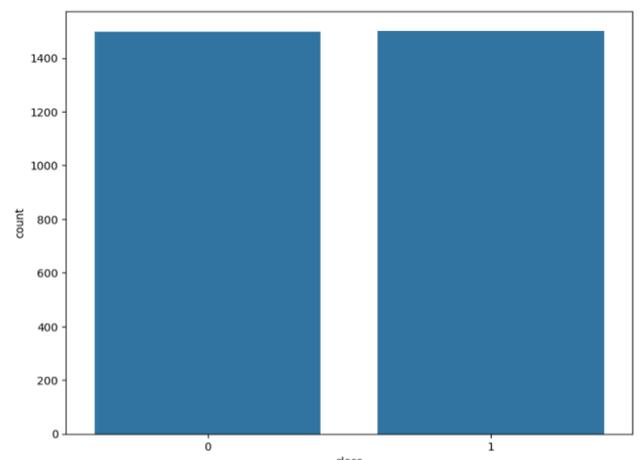
Penelitian ini bertujuan untuk mengembangkan model klasifikasi gambar menggunakan ResNet-34 untuk klasifikasi wajah *Down Syndrome* dengan akurasi tinggi. Tahapan penelitian yang akan dilakukan ditunjukkan pada Gambar 1. Metode penelitian menjelaskan pendekatan, rancangan kegiatan, ruang lingkup atau objek, bahan dan alat utama, tempat, teknik pengumpulan data, definisi operasional variabel penelitian, dan teknik analisis.



Gambar 1. Metodologi Penelitian

**A. Pengumpulan Data**

Langkah pertama adalah pengumpulan data dimana data yang diambil berasal dari kaggle dengan jumlah 3000 gambar yang terdiri dari 2 kelas data antara lain kelas *down syndrome* dan *healthy*, setelah itu dilakukan pengecekan *imbalance data* didapatkan data sudah *balance* antara 2 kelas tersebut dapat dilihat pada gambar 2, kemudian data tersebut dibagi ke dalam 3 kategori data antara lain data *training*, *validation* dan *test*, detail jumlah gambar dapat dilihat pada Tabel I.



Gambar 2. Pengecekan Imbalance Data

TABEL I.  
DETAIL JENIS DATA YANG DIGUNAKAN

Tipe Data	Kelas	
	Down Syndrome (Jumlah)	Healthy (Jumlah)
Training	1046	1053
Validation	307	297
Testing	147	150

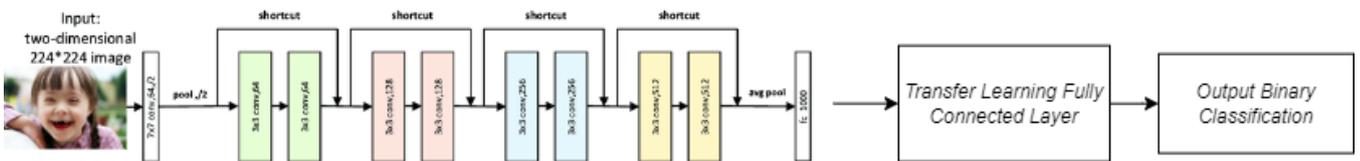
**B. Augmentasi Data**

Tahapan augmentasi data adalah tahapan untuk memodifikasi gambar untuk menambahkan variasi dataset menjadi lebih beragam dimana tahapan augmentasi yang dilakukan antara lain adalah *resize* ke ukuran 224 X 224 dan *random resize crop* ke ukuran 250 juga dan melakukan *random horizontal flip*. Dataset penelitian ini terdiri dari 3.000 gambar, terbagi rata antara kelas *Down Syndrome* dan Sehat, dan diambil dari platform Kaggle. Dengan ukuran dataset yang relatif kecil, augmentasi data sangat penting untuk meningkatkan variasi dan kualitas data yang digunakan dalam pelatihan model. Teknik *random cropping* dan *horizontal flipping* diterapkan untuk menciptakan variasi visual yang lebih banyak dari gambar yang ada. Teknik augmentasi ini bertujuan untuk meningkatkan generalisasi model dengan mencegah *overfitting* dan memungkinkan model belajar dari berbagai variasi wajah penderita *Down Syndrome*.

**C. Training Model**

Pada tahap ini, model arsitektur dikembangkan dengan menggunakan algoritma CNN, memanfaatkan model ResNet-34 sebagai model *pre-trained*. Gambar yang awalnya dalam format tiga dimensi kemudian diubah menjadi satu dimensi menggunakan proses *flattening* agar bisa dilatih menggunakan *fully connected layer*. Hasil akhir dari proses ini berupa klasifikasi gambar.

Transfer learning melibatkan *fine-tuning* model yang telah dilatih sebelumnya. Dalam penelitian ini, layer awal dari ResNet34 yang bertanggung jawab untuk menangkap fitur umum (seperti tepi, tekstur, dan bentuk dasar) dibekukan. Dengan membekukan layer ini, model dapat mempertahankan pengetahuan yang telah diperolehnya dari dataset ImageNet, sementara layer-layer terakhir diubah dan dilatih ulang untuk menangani tugas klasifikasi yang lebih spesifik, yaitu klasifikasi wajah *Down Syndrome*. Proses *fine-tuning* ini jauh lebih efisien daripada pelatihan dari awal, dan memungkinkan model untuk mencapai hasil yang baik meskipun dataset yang digunakan berukuran terbatas. Arsitektur yang digunakan dapat dilihat pada gambar 3.

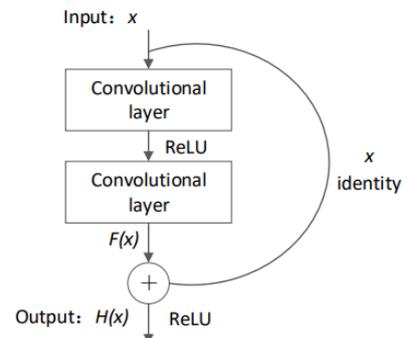


Gambar 3. Arsitektur ResNet dengan Transfer Learning

Dalam penelitian ini, pendekatan transfer learning dipilih karena lebih efisien dalam hal waktu dan sumber daya komputasi, terutama mengingat ukuran dataset yang terbatas. Dengan memanfaatkan ResNet34 yang telah di-*pretrain* pada dataset besar seperti ImageNet, kami dapat memanfaatkan fitur-fitur umum yang telah dipelajari sebelumnya. Layer awal model yang menangkap fitur dasar dibekukan, dan layer-

layer terakhir diubah untuk tugas spesifik, yaitu klasifikasi wajah *Down Syndrome*. Pendekatan ini memungkinkan model mencapai akurasi yang tinggi tanpa perlu memulai pelatihan dari awal.

ResNet-34 menggunakan blok residual, di mana setiap blok residual terdiri dari dua atau lebih lapisan konvolusi. Output blok ini diperoleh dengan menambahkan input blok ke output lapisan terakhir dalam blok tersebut. Strategi ini memungkinkan jaringan untuk mempelajari representasi residual, yang berarti jaringan belajar menghitung perbedaan antara input dan output yang diinginkan, daripada berusaha mendekati output secara langsung. Blok residual dijelaskan lebih lanjut pada Gambar 4.



Gambar 4. Blok Residual pada Arsitektur ResNet

Dimana perhitungan matematis untuk residual blok disajikan dalam persamaan 1.

$$H(x) = F(x) + x \tag{1}$$

SoftMax diperkenalkan sebagai aktivasi untuk meramalkan tergantung pada likelihood maksimum sehingga dapat digunakan dalam mengklasifikasi gambar multi kelas karena dapat menentukan probabilitas prediksi pada masing-masing kelas. Persamaan untuk fungsi SoftMax diberikan pada persamaan 2.

$$S^{e,j} = \frac{e^{z_{i,j}}}{\sum_{l=1}^L e^{z_{i,l}}} \tag{2}$$

AdamW adalah algoritma optimisasi yang sering digunakan dalam pelatihan jaringan saraf tiruan. Algoritma ini merupakan modifikasi dari optimisasi Adam, yang menambahkan komponen *weight decay* secara eksplisit. Pada

Adam, regularisasi L2 yang biasanya diterapkan tidak bekerja secara optimal karena adanya pembaruan gradien. Namun, AdamW memisahkan *weight decay* dari gradien *update*, sehingga memberikan regularisasi yang lebih efektif. Pada AdamW, pembaruan parameter dilakukan menggunakan persamaan 3.

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{m_t}{\sqrt{v_t + \epsilon}} - \eta \cdot \lambda \cdot \theta_t \quad (3)$$

Rumus ini terdiri dari dua bagian utama yaitu persamaan 4.

$$\eta \cdot \frac{m_t}{\sqrt{v_t + \epsilon}} \quad (4)$$

Bagian ini adalah mekanisme pembaruan parameter seperti pada Adam. Di sini,  $m_t$  merupakan rata-rata momentum dari gradien, dan  $v_t$  adalah estimasi variansi dari gradien. Rata-rata momentum  $m_t$  memberikan kontribusi untuk menjaga jejak gradien masa lalu, sementara  $v_t$  menormalkan pembaruan menggunakan akar dari rata-rata kuadrat gradien, yang bertujuan untuk mengurangi ketidakstabilan numerik. Faktor  $\eta$  adalah *learning rate* yang mengontrol ukuran langkah pembaruan parameter yang dijelaskan pada persamaan 5.

$$-\eta \cdot \lambda \cdot \theta_t \quad (5)$$

Bagian ini memperkenalkan *decay* eksplisit dari parameter  $\theta_t$ , yang dikontrol oleh faktor  $\lambda$ . *Weight decay* ini memaksa parameter model mendekati nol, yang membantu mencegah *overfitting* dengan mengurangi kompleksitas model. Tidak seperti pada Adam yang menggunakan regularisasi L2, AdamW mengimplementasikan *weight decay* secara terpisah, sehingga *decay* ini diterapkan secara independen dari pembaruan gradien, yang membuatnya lebih efektif dalam menjaga kestabilan optimisasi.

Menurut Loshchilov dan Hutter [10], pembaruan ini membantu model generalisasi lebih baik pada data baru, terutama dalam konteks *deep learning*, di mana model cenderung mengalami *overfitting* jika regularisasi yang tepat tidak diterapkan.

Pada tahap *training* pada *fully connected layer* data akan menggunakan data *training* akan menggunakan data *training* yang telah dibagi pada tahap sebelumnya dimana proses *training* di jalankan di atas mesin GPU CUDA A100 dengan menggunakan *epoch* 100 dan dilakukan 3 kali percobaan dengan *hyperparameter* yang berbeda seperti yang ditunjukkan pada Tabel II.

TABEL II.  
DETAIL HYPERPARAMETER

Percobaan	Hyperparameter	
	Batch size	Learning Rate
Percobaan 1	32	0.0001
Percobaan 2	128	0.001
Percobaan 3	256	0.001

Dalam penelitian ini, dilakukan pemilihan *hyperparameter* secara teliti untuk mengeksplorasi pengaruh variasi *batch*

*size*, *learning rate*, dan jumlah *epochs* terhadap kinerja model ResNet34. Pemilihan *hyperparameter* yang tepat sangat penting dalam melatih model *deep learning* agar mencapai performa yang optimal tanpa mengalami masalah *overfitting* atau *underfitting*[7]. Oleh karena itu, tiga percobaan dengan konfigurasi *hyperparameter* yang berbeda dilakukan guna menemukan konfigurasi yang paling sesuai untuk tugas klasifikasi wajah *Down Syndrome*. *Batch size* merujuk pada jumlah sampel yang diproses sebelum model melakukan pembaruan bobot. Dalam penelitian ini, variasi *batch size* dilakukan untuk mengevaluasi bagaimana ukuran *batch* mempengaruhi proses pelatihan.

Pada percobaan pertama, digunakan *batch size* yang kecil, yaitu 32, yang memungkinkan model untuk belajar lebih rinci pada setiap iterasi. *Batch size* yang kecil sering kali memberikan pembaruan bobot yang lebih sering, sehingga memungkinkan model fokus pada detail-detail kecil dari data. Namun, hal ini juga dapat meningkatkan risiko *overfitting*, di mana model terlalu menyesuaikan diri dengan data *training* dan kesulitan melakukan generalisasi pada data yang belum pernah dilihat sebelumnya.

Pada percobaan kedua, digunakan *batch size* yang lebih besar, yaitu 128, yang bertujuan untuk meningkatkan efisiensi proses pelatihan. Dengan *batch size* yang lebih besar, model mendapatkan lebih banyak sampel pada setiap iterasi, yang memungkinkan model untuk menangkap pola yang lebih umum dan tidak terlalu fokus pada detail kecil yang mungkin hanya muncul pada data *training*. Penggunaan *batch size* yang lebih besar ini diharapkan dapat meningkatkan kemampuan model dalam melakukan generalisasi pada data baru yang tidak terlihat selama pelatihan.

Pada percobaan ketiga, digunakan *batch size* yang sangat besar, yaitu 256. Penggunaan *batch size* yang lebih besar memungkinkan model untuk mempercepat proses pelatihan karena lebih banyak data diproses pada setiap iterasi. Namun, *batch size* yang sangat besar juga memiliki kelemahan, yaitu model mungkin kehilangan kemampuan untuk menangkap pola-pola yang lebih halus dari data. Hal ini dapat menyebabkan *underfitting*, di mana model tidak cukup belajar dari data dan performa pada data *validation* menurun.

Percobaan ini bertujuan untuk melihat apakah peningkatan kecepatan pelatihan dapat mengimbangi risiko penurunan generalisasi. Selain *batch size*, *learning rate* juga merupakan *hyperparameter* penting yang mengatur seberapa besar perubahan yang dilakukan pada bobot model selama pelatihan.

Pada percobaan pertama, digunakan *learning rate* yang rendah, yaitu 0.0001. *Learning rate* yang rendah ini dipilih untuk memastikan bahwa perubahan bobot terjadi secara perlahan, yang membantu mencegah model melompat terlalu jauh pada solusi yang salah. Namun, *learning rate* yang rendah juga membuat proses pelatihan berjalan lebih lambat, yang dapat menyebabkan model memerlukan waktu yang lebih lama untuk mencapai konvergensi.

Pada percobaan kedua dan ketiga, digunakan *learning rate* yang lebih tinggi, yaitu 0.001. *Learning rate* yang lebih tinggi

memungkinkan model belajar lebih cepat dan mencapai konvergensi dengan lebih efisien. Namun, *learning rate* yang lebih tinggi ini tetap dipilih dengan hati-hati agar tidak menyebabkan model melompat terlalu jauh dari nilai optimal, yang dapat menyebabkan performa model menurun. Dalam percobaan ini, *learning rate* yang lebih tinggi diharapkan dapat mempercepat proses pelatihan tanpa mengorbankan akurasi.

Jumlah *epochs* juga merupakan komponen penting dalam menentukan seberapa lama model dilatih. Pada penelitian ini, jumlah *epochs* disesuaikan berdasarkan performa yang diperoleh selama pelatihan dan dengan memonitor akurasi pada data *training* serta *validation*. Dengan pemilihan *hyperparameter* yang tepat, penelitian ini bertujuan untuk menemukan kombinasi yang memberikan keseimbangan antara kecepatan pelatihan dan kemampuan model untuk melakukan generalisasi yang baik terhadap data yang belum pernah dilihat sebelumnya.

#### D. Evaluasi

Evaluasi dilakukan menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1 Score* yang didasarkan pada *Confusion Matrix* setelah pengujian dilakukan pada data *testing*. Hasil dari evaluasi ini akan menjadi dasar dalam menentukan kelayakan metode yang digunakan dalam penelitian. Setiap metrik memiliki peran penting dalam mengukur performa model. *Accuracy* mengukur persentase prediksi yang benar, *Precision* mengukur proporsi prediksi positif yang benar, *Recall* mengukur seberapa baik model mendeteksi sampel positif yang sebenarnya, dan *F1 Score* adalah rata-rata harmonis dari *Precision* dan *Recall*, yang sangat berguna ketika ada ketidakseimbangan data [11]. Berikut adalah rumus masing-masing metrik.

*Accuracy*: Mengukur seberapa sering model melakukan prediksi yang benar. Rumus *Accuracy* ditunjukkan pada persamaan 6.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Di mana: *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, *FN* = False Negatives. *Precision*: Mengukur akurasi prediksi positif. Rumus *Precision* ditunjukkan pada persamaan 7.

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

*Recall*: Mengukur seberapa baik model mendeteksi kelas positif. Rumus *recall* ditunjukkan pada persamaan 8.

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

*F1 Score*: Rata-rata harmonis dari *Precision* dan *Recall*. Rumus *F1 Score* ditunjukkan pada persamaan 9.

$$F1Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (9)$$

Evaluasi berdasarkan metrik-metrik ini memberikan gambaran komprehensif tentang performa model dari berbagai aspek.

*Accuracy* cocok digunakan ketika jumlah data seimbang, sementara *Precision*, *Recall*, dan *F1 Score* lebih cocok untuk data yang tidak seimbang. Penggunaan metrik yang tepat memastikan penilaian yang lebih akurat terhadap kelayakan model dalam menangani masalah klasifikasi.

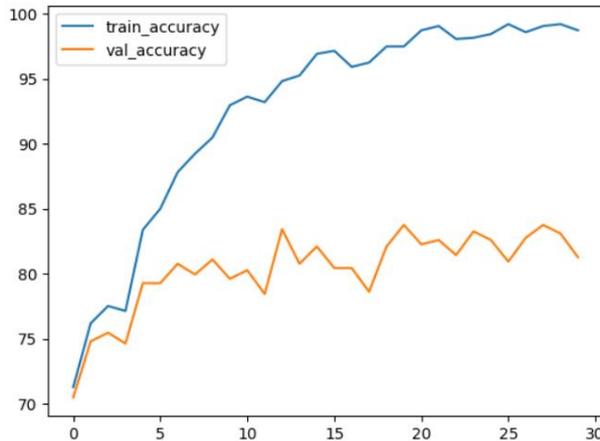
Dalam konteks ini, tujuan utama adalah untuk meminimalkan jumlah *false negatives*, karena hal ini berpotensi menyebabkan penderita *Down Syndrome* tidak mendapatkan penanganan yang tepat. *False negatives* terjadi ketika individu yang sebenarnya positif tidak terdeteksi oleh model, yang dalam konteks medis dapat berakibat serius karena keterlambatan atau kegagalan dalam diagnosis. Oleh karena itu, *recall* yang tinggi lebih diutamakan, meskipun terkadang ini dapat menyebabkan peningkatan *false positives* (yaitu individu sehat yang secara salah diklasifikasikan sebagai *Down Syndrome*). Dalam skenario medis, *false positives* mungkin kurang berbahaya dibandingkan *false negatives*, karena dapat dikoreksi dengan tes lanjutan, sementara *false negatives* bisa mengakibatkan masalah serius dalam jangka panjang.

Sebagai contoh, penelitian oleh Powers [12] menunjukkan bahwa dalam situasi di mana terlewatnya diagnosis dapat membahayakan, seperti dalam deteksi penyakit, *recall* yang tinggi sangat penting untuk memastikan bahwa semua kasus positif teridentifikasi. Studi oleh Sokolova dan Lapalme [13] juga menguatkan hal ini dengan menekankan bahwa *recall* adalah metrik yang lebih relevan ketika tujuan utama adalah memastikan bahwa semua individu dengan kondisi tertentu terdeteksi, terutama dalam situasi di mana *false negatives* dapat menyebabkan konsekuensi buruk. Sebagai tambahan, Steyerberg et al. [14] menunjukkan bahwa dalam konteks klinis, menghindari *false negatives* sangat penting untuk diagnosis dini dan intervensi tepat waktu, yang dapat berdampak besar pada hasil kesehatan pasien. Oleh karena itu, dalam deteksi wajah *Down Syndrome* [15-17], *recall* yang tinggi lebih diutamakan untuk memastikan bahwa individu yang membutuhkan penanganan dapat segera didiagnosis, meskipun ini dapat meningkatkan jumlah *false positives* yang kemudian dapat diperbaiki dengan pengujian lebih lanjut.

### III. HASIL DAN PEMBAHASAN

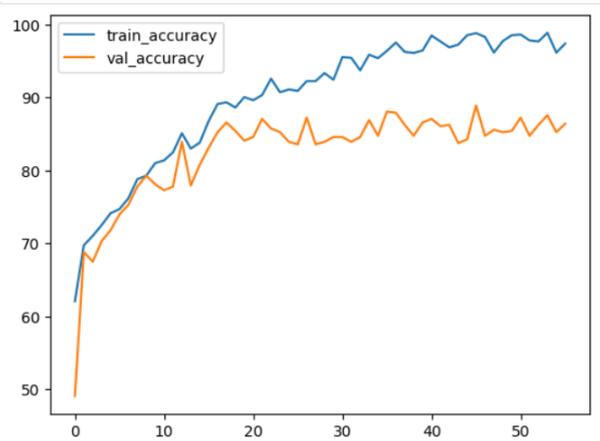
Hasil pengujian dari implementasi pada percobaan pertama yaitu *batch size* 32 dan *learning rate* 0.0001 pada proses *training* didapatkan hasil *accuracy* 98.7 % pada data *training* dan 81.2% pada data *validation* dan berhenti pada *epoch* ke 30 dimana *accuracy* pada data *training* dan *validation* disajikan pada Gambar 5, dimana terlihat sangat *overfit*. Pada diagram *accuracy* percobaan pertama, kita melihat bahwa akurasi *training* mencapai nilai yang sangat tinggi dengan cepat, menunjukkan bahwa model mampu belajar dengan baik dari data *training*. Namun, akurasi pada data *validation* jauh lebih rendah dibandingkan data *training*, yang

menunjukkan bahwa model mengalami *overfitting*. *Overfitting* terjadi karena model terlalu menyesuaikan diri dengan data *training* dan gagal melakukan generalisasi pada data baru yang tidak pernah dilihat sebelumnya. Ini disebabkan oleh *batch size* yang kecil, yang menyebabkan model mempelajari detail kecil pada data *training*, serta *learning rate* yang rendah, yang membuat proses pelatihan berjalan lambat tanpa cukup memperbaiki generalisasi model.



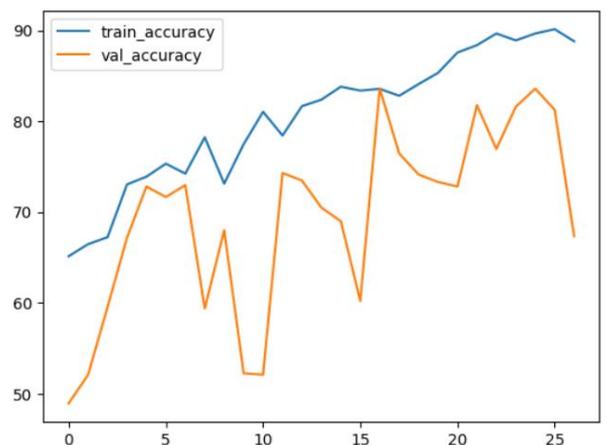
Gambar 5. Grafik Accuracy Percobaan 1

Pada percobaan kedua yaitu *batch size* 128 dan *learning rate* 0.001 pada proses *training* didapatkan hasil *accuracy* 97.37 % pada data *training* dan 86.40% pada data *validation* dan berhenti pada *epoch* ke 56 dimana *accuracy* pada data *training* dan *validation* disajikan pada Gambar 6, dimana terlihat masih *overfit* namun ada peningkatan pada *accuracy* data *validation*. Pada percobaan kedua, diagram *accuracy* menunjukkan peningkatan yang lebih stabil baik pada data *training* maupun data *validation*. Akurasi *training* masih meningkat, namun dengan laju yang lebih wajar, dan akurasi *validation* juga meningkat secara signifikan mendekati akurasi *training*. Ini menunjukkan bahwa model memiliki kemampuan generalisasi yang lebih baik dan tidak *overfit* pada data *training*. Kombinasi *batch size* yang lebih besar dan *learning rate* yang lebih tinggi membuat model mampu belajar lebih cepat dan menangkap pola yang lebih umum, sehingga meningkatkan performa pada data *validation*. Percobaan ini menghasilkan akurasi *validation* terbaik di antara ketiga percobaan.



Gambar 6. Grafik Accuracy Percobaan 2

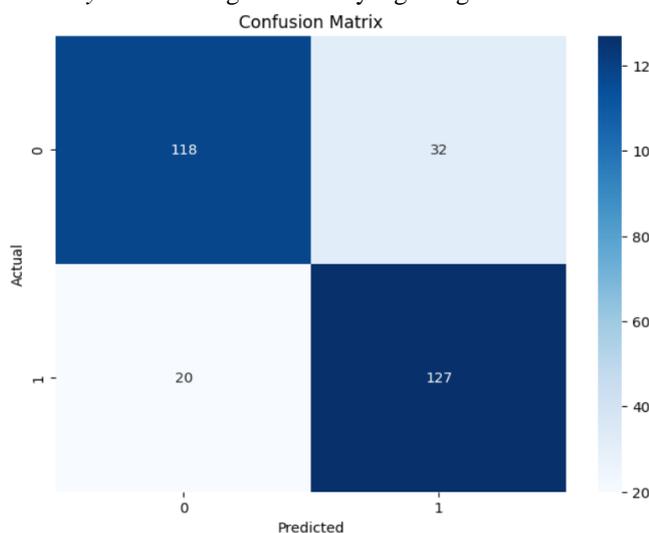
Pada percobaan ketiga yaitu *batch size* 256 dan *learning rate* 0.001 pada proses *training* didapatkan hasil *accuracy* 88.80 % pada data *training* dan 67.33% pada data *validation* dan berhenti pada *epoch* ke 58 dimana *accuracy* pada data *training* dan *validation* disajikan pada Gambar 7, dimana terlihat *overfit* namun dan terdapat penurunan pada *accuracy* data *validation*. Pada percobaan ketiga, diagram *accuracy* menunjukkan penurunan yang signifikan pada akurasi *validation*, meskipun akurasi *training* tetap tinggi. Ini menunjukkan bahwa model mengalami *underfitting*. *Batch size* yang terlalu besar membuat model kehilangan kemampuan untuk menangkap pola yang lebih halus dalam data, sehingga gagal melakukan generalisasi dengan baik pada data *validation*. Meskipun *learning rate* tetap tinggi dan mempercepat pelatihan, *batch size* yang terlalu besar membuat model belajar dari pola yang terlalu umum, sehingga performa pada data *validation* turun drastis.



Gambar 7. Grafik Accuracy Percobaan 3

Dari 3 percobaan tersebut akan dibentuk *confusion matrix* untuk melihat hasil evaluasi pada data *testing*. Pada *confusion matrix* tersebut nilai 0 dan 1 melambangkan jenis kelas dimana 0 merupakan *healthy* dan 1 adalah *down*

*syndrome*. Pada *confussion matrix* percobaan pertama terlihat data *healthy* yang berhasil diprediksi adalah 118 dan yang gagal di prediksi 32, sedangkan data *down syndrome* yang berhasil diprediksi sebanyak 127 dan gagal diprediksi sebanyak 20. *Confussion matrix* percobaan pertama dapat dilihat pada Gambar 8. Pada percobaan pertama, *confussion matrix* menunjukkan bahwa model melakukan prediksi yang cukup baik pada kelas *down syndrome* dan *healthy*, namun ada ketidakseimbangan yang terlihat dalam jumlah kesalahan prediksi. Model mampu memprediksi sebagian besar data dengan benar, tetapi ada jumlah *false negatives* yang signifikan (penderita *Down Syndrome* yang diklasifikasikan sebagai Sehat). Hal ini disebabkan oleh *overfitting*, di mana model terlalu menyesuaikan diri pada data *training* dan tidak dapat melakukan generalisasi dengan baik pada data *validation*. Sehingga, model gagal mendeteksi beberapa kasus *Down Syndrome* dengan akurasi yang diinginkan.



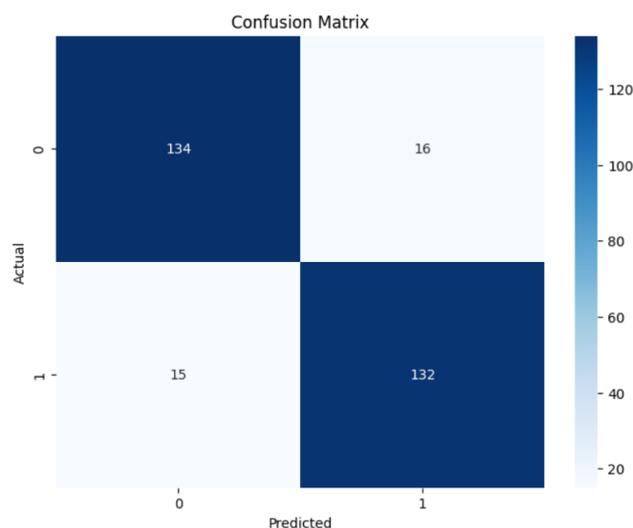
Gambar 8. *Confussion Matrix* Percobaan 1

Pada *confussion matrix* percobaan kedua terlihat data *healthy* yang berhasil diprediksi adalah 134 dan yang gagal di prediksi 16, sedangkan data *down syndrome* yang berhasil diprediksi sebanyak 132 dan gagal diprediksi sebanyak 15. *Confussion matrix* percobaan pertama dapat dilihat pada Gambar 9. Pada percobaan kedua, *confussion matrix* menunjukkan hasil yang paling seimbang di antara ketiga percobaan. Model berhasil mendeteksi sebagian besar kasus *down syndrome* dan Sehat dengan baik, dengan jumlah *false positives* dan *false negatives* yang lebih rendah dibandingkan percobaan pertama. *True Positives* (TP) dan *True Negatives* (TN) meningkat, yang menunjukkan bahwa model mampu menangkap pola-pola penting dalam data dan melakukan generalisasi dengan baik. Jumlah *false negatives* berkurang, yang berarti model lebih jarang gagal mendeteksi penderita *Down Syndrome*.

Keberhasilan percobaan ini dalam mengurangi *false negatives* penting, terutama dalam konteks medis, karena kesalahan dalam mendeteksi kasus positif (*Down Syndrome*) memiliki konsekuensi serius. Model pada percobaan ini

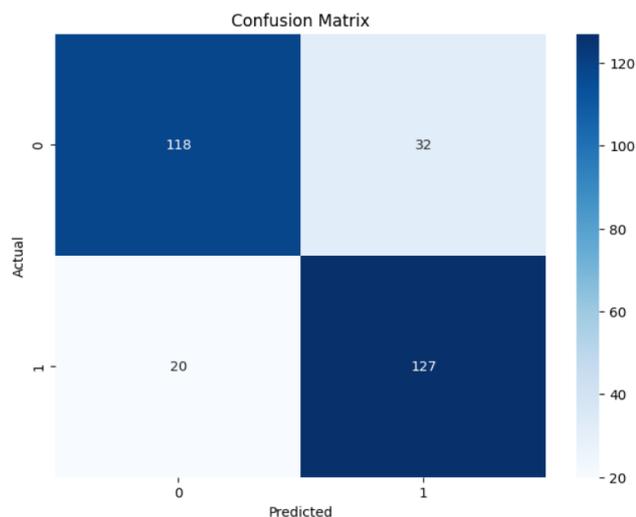
menunjukkan keseimbangan yang baik antara mendeteksi penderita *Down Syndrome* dan tidak salah mendeteksi individu sehat sebagai *Down Syndrome*.

Pada *confussion matrix* percobaan ketiga terlihat data *healthy* yang berhasil diprediksi adalah 118 dan yang gagal di prediksi 32, sedangkan data *down syndrome* yang berhasil diprediksi sebanyak 127 dan gagal diprediksi sebanyak 20. *Confussion matrix* percobaan pertama dapat dilihat pada Gambar 10. Pada percobaan ketiga, *confussion matrix* menunjukkan adanya penurunan kinerja model, khususnya dalam mendeteksi kasus *Down Syndrome*. *False negatives* meningkat secara signifikan, yang berarti banyak penderita *Down Syndrome* yang salah diklasifikasikan sebagai Sehat. Hal ini menunjukkan bahwa model mengalami *underfitting*, di mana model tidak belajar cukup dari data *training* dan gagal menangkap pola-pola yang lebih spesifik dalam data. Jumlah *false positives* juga sedikit meningkat, tetapi masalah utama adalah *false negatives*, di mana model gagal mendeteksi banyak kasus *Down Syndrome*.



Gambar 9. *Confussion Matrix* Percobaan 2

*True Positives* (TP) dan *True Negatives* (TN) lebih rendah dibandingkan percobaan kedua, yang mengindikasikan bahwa model tidak mampu melakukan generalisasi dengan baik pada data *validation*, meskipun akurasi *training* cukup tinggi. *Batch size* yang terlalu besar membuat model tidak dapat mempelajari detail penting dari data, sehingga kinerja model dalam mendeteksi kasus yang sebenarnya positif (*Down Syndrome*) menurun drastis.



Gambar 10. Confusion Matrix Percobaan

Dari confusion matrix akan dihitung nilai accuracy, precision, recall dan F1-score pada masing-masing percobaan. Dalam percobaan 1 menghasilkan akurasi 82.83%, precision 0.8362, recall 0.8350, dan F1 score 0.8348, menunjukkan kinerja yang cukup baik namun masih di bawah 85% akurasi. Percobaan 2 memberikan hasil terbaik dengan akurasi 87.88%, precision 0.8956, recall 0.8956, dan F1 score 0.8956, menandakan model ini lebih unggul dalam keseimbangan antara prediksi benar dan kesalahan. Sementara itu, percobaan 3 memiliki akurasi terendah sebesar 80.47%, dengan precision 0.8272, recall 0.8249, dan F1 score 0.8247, menunjukkan kinerja yang lebih rendah dibandingkan percobaan lainnya. Hasil dari 3 percobaan dapat dilihat pada Tabel III.

Dari hasil tiga percobaan yang dilakukan, dapat disimpulkan bahwa pemilihan batch size dan learning rate memiliki dampak signifikan terhadap kinerja model dalam melakukan klasifikasi wajah Down Syndrome. Percobaan kedua, yang menggunakan batch size 128 dan learning rate 0.001, menunjukkan hasil terbaik dengan akurasi yang tinggi pada data validation. Model pada percobaan ini mampu mencapai keseimbangan yang baik antara precision dan recall, yang menghasilkan F1-score tertinggi di antara semua percobaan. Hal ini menunjukkan bahwa model tidak hanya mampu mendeteksi individu dengan Down Syndrome (recall yang tinggi) tetapi juga mengurangi jumlah kesalahan positif (false positives), yang tercermin dalam nilai precision yang baik.

Sebaliknya, pada percobaan pertama dengan batch size yang kecil (32) dan learning rate yang rendah (0.0001), meskipun akurasi training tinggi, model mengalami overfitting yang signifikan. Ini terlihat dari rendahnya akurasi pada data validation, serta nilai recall yang rendah, yang berarti model gagal mendeteksi banyak kasus Down Syndrome (false negatives yang tinggi). Hal ini menunjukkan bahwa model terlalu menyesuaikan diri dengan data training

dan tidak mampu melakukan generalisasi yang baik pada data baru.

Pada percobaan ketiga, dengan batch size 256 dan learning rate 0.001, model mengalami underfitting. Akurasi validation menurun drastis, dan jumlah false negatives meningkat secara signifikan, yang ditunjukkan oleh rendahnya nilai recall. Ini menunjukkan bahwa batch size yang terlalu besar membuat model kehilangan kemampuan untuk menangkap pola-pola penting dalam data, sehingga tidak mampu mendeteksi kasus Down Syndrome secara akurat.

TABEL III.  
HASIL PERCOBAAN

Percobaan	Accuracy	Precision	Recall	F1 Score
Percobaan 1	82.83	0.8362	0.835	0.8348
Percobaan 2	87.88	0.8956	0.8956	0.8956
Percobaan 3	80.47	0.8272	0.8249	0.8247

Secara keseluruhan, percobaan kedua memberikan hasil yang paling optimal, dengan nilai akurasi, precision, recall, dan F1-score yang lebih seimbang. Percobaan ini menunjukkan bahwa batch size 128 dan learning rate 0.001 merupakan kombinasi yang tepat untuk memaksimalkan performa model dalam tugas klasifikasi wajah Down Syndrome, menghindari masalah overfitting dan underfitting yang muncul pada percobaan pertama dan ketiga.

#### IV. KESIMPULAN

Penelitian ini telah mengeksplorasi penggunaan arsitektur ResNet34 dengan metode transfer learning untuk klasifikasi wajah penderita Down Syndrome. Melalui tiga percobaan dengan variasi batch size dan learning rate, hasil menunjukkan bahwa pemilihan hyperparameter yang tepat sangat memengaruhi performa model dalam mendeteksi wajah Down Syndrome secara akurat.

Percobaan kedua, yang menggunakan batch size 128 dan learning rate 0.001, memberikan hasil yang paling optimal, dengan akurasi tertinggi pada data validation, serta keseimbangan yang baik antara precision, recall, dan F1-score. Model dalam percobaan ini mampu menghindari masalah overfitting dan underfitting, yang terjadi pada percobaan pertama dan ketiga. Percobaan pertama dengan batch size kecil (32) mengalami overfitting, sementara percobaan ketiga dengan batch size besar (256) menunjukkan underfitting yang mengakibatkan rendahnya akurasi pada data validation.

Kesimpulan dari penelitian ini adalah bahwa batch size 128 dan learning rate 0.001 merupakan kombinasi terbaik untuk meningkatkan kemampuan model dalam melakukan generalisasi dan mendeteksi kasus Down Syndrome secara akurat. Selain itu, nilai recall yang tinggi pada percobaan kedua menunjukkan bahwa model mampu mendeteksi sebagian besar penderita Down Syndrome, yang penting dalam konteks medis untuk meminimalkan jumlah false negatives. Dengan demikian, model ini dapat membantu dalam diagnosis dini, yang merupakan faktor krusial dalam

penanganan dan perawatan yang tepat bagi penderita *Down Syndrome*.

Penelitian ini membuka peluang untuk eksplorasi lebih lanjut dalam meningkatkan performa model, seperti menggunakan *dataset* yang lebih besar dan menerapkan teknik augmentasi data yang lebih beragam. Selain itu, pendekatan yang digunakan dapat diterapkan pada tugas klasifikasi lain dalam domain medis, khususnya dalam mendeteksi kondisi atau kelainan yang membutuhkan deteksi dini dan akurasi yang tinggi.

#### UCAPAN TERIMA KASIH

Penelitian ini didukung oleh dana hibah penelitian kompetitif yang diberikan oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemdikbudristek) melalui skema Penelitian Dosen Pemula (PDP). Kami mengucapkan terima kasih yang sebesar-besarnya kepada Direktorat Jenderal Pendidikan Tinggi (DIKTI) atas dukungan dan kesempatan yang diberikan untuk melaksanakan penelitian ini.

#### DAFTAR PUSTAKA

- [1] K. He, X. Zhang, S. Ren, dan J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770-778.
- [2] J. Yosinski, J. Clune, Y. Bengio, dan H. Lipson, "How transferable are features in deep neural networks?," *Advances in neural information processing systems*, vol. 27, pp. 3320-3328, 2014.
- [3] G. Huang, Z. Liu, L. van der Maaten, dan K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2261-2269.
- [4] Z. Zhu, X. Huang, dan D. Du, "ResNet-based deep learning model for medical image analysis," *J. Med. Imag. Health Inform.*, vol. 9, no. 5, pp. 1023-1030, 2019.
- [5] S. J. Pan, dan Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345-1359, 2019.
- [6] G. Voulodimos, N. Doulamis, A. Doulamis, dan E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1-13, 2018.
- [7] J. Smith, "A disciplined approach to neural network hyper-parameters: Part 1—learning rate, batch size, momentum, and weight decay," *arXiv preprint arXiv:1803.09820*, 2018.
- [8] F. Rozaan, "Identifikasi wajah pengidap Down Syndrome menggunakan model CNN," *Makalah IF4073 Interpretasi dan Pengolahan Citra*, Institut Teknologi Bandung, 2023(Makalah-IF4073-Citra-20...).
- [9] M. Cayli, "Detection of Down Syndrome in Children," *Kaggle*, 2023. [Online]. Available: <https://www.kaggle.com/datasets/mervecayli/detection-of-down-syndrome-in-children/data>. [Accessed: 01-Aug-2024].
- [10] I. Loshchilov dan F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2019.
- [11] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica*, vol. 31, no. 3, pp. 249-268, 2007.
- [12] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37-63, 2011.
- [13] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427-437, Jul. 2009.
- [14] E. W. Steyerberg, *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. New York, NY, USA: Springer Science & Business Media, 2010.
- [15] B. Jin, L. Cruz and N. Gonçalves, "Deep Facial Diagnosis: Deep Transfer Learning From Face Recognition to Facial Diagnosis," in *IEEE Access*, vol. 8, pp. 123649-123661, 2020, doi: 10.1109/ACCESS.2020.3005687.
- [16] Ş. Kiliç, İ. Askerzade and Y. Kaya, "Using ResNet Transfer Deep Learning Methods in Person Identification According to Physical Actions," in *IEEE Access*, vol. 8, pp. 220364-220373, 2020, doi: 10.1109/ACCESS.2020.3040649.
- [17] J. Wang et al., "Multiple Genetic Syndromes Recognition Based on a Deep Learning Framework and Cross-Loss Training," in *IEEE Access*, vol. 10, pp. 117084-117092, 2022, doi: 10.1109/ACCESS.2022.3218160.