

Evaluasi Ensemble Stacking Arsitektur EfficientNetB3 dan EfficientNetV2S (Studi Kasus Klasifikasi Penyakit Alzheimer)

GALIH ASHARI RAKHMAT^{1*}, FIRLY TAUFIKUROHMAN¹

¹Program Studi Informatika, Institut Teknologi Nasional Bandung
galih@itenas.ac.id

Received 04 09 2023 | Revised 11 09 2023 | Accepted 11 09 2023

ABSTRAK

Perkembangan artificial intelligence pada bidang deep learning pada computer vision banyak diterapkan dalam mendeteksi penyakit salah satunya Alzheimer. Selain dibutuhkan suatu model dengan akurasi tinggi juga dibutuhkan efisiensi. Penelitian ini bertujuan untuk menggabungkan hasil prediksi EfficientNetB3 dengan EfficientNetV2S untuk meningkatkan accuracy dan menurunkan tingkat kesalahan. Metode ensemble stacking menggabungkan prediksi dari kedua model untuk meningkatkan kinerja model. Hasil yang diperoleh model kinerja tertinggi berdasarkan metrik evaluasi diperoleh pada ensemble model pada learning rate 0,0001 pada 50 epoch dengan evaluasi performa model 99,98% pada accuracy, 99,98% precision, 99,98% recall, dan 99,98% f1-score pada pengujian 6400 data. Selain itu ensemble model dapat menurunkan kesalahan prediksi yaitu total 1 kesalahan.

Kata kunci: *Deep Learning, EfficientNetB3, EfficientNetV2S, Ensemble Stacking, Alzheimer*

ABSTRACT

The advancement of artificial intelligence in the field of deep learning, particularly in computer vision, has been widely applied in disease detection, including Alzheimer's disease. Beyond requiring a highly accurate model, efficiency is also crucial. This study aims to combine the prediction outcomes of EfficientNetB3 and EfficientNetV2S to enhance accuracy and reduce error rates. The ensemble stacking method merges predictions from both models to enhance overall performance. The best performing model's results were achieved using an ensemble model with a learning rate of 0.0001, over 50 epochs. The model demonstrated an evaluation performance of 99.98% accuracy, 99.98% precision, 99.98% recall, and 99.98% F1-score in testing with 6400 data points. Furthermore, the ensemble model managed to reduce prediction errors to a total of 1 mistakes.

Keywords: *Deep Learning, EfficientNetB3, EfficientNetV2S, Ensemble Stacking, Alzheimer's*

1. PENDAHULUAN

Deep learning merupakan metode *learning* yang memanfaatkan *artificial neural network* yang berlapis-lapis (*multilayer*). *Artificial Neural Network* ini dibuat mirip dengan otak manusia, dimana neuron-neuron terkoneksi satu sama lain sehingga membentuk sebuah jaringan neuron yang sangat rumit. *Deep learning* atau *structural learning* atau *hirarchial learning* atau *deep neural* merupakan metode *learning* yang memanfaatkan *multiple non-linear transformation*, *deep learning* dapat dipandang sebagai gabungan *machine learning* dengan *AI* (*Artificial neural network*) (Adi Nugroho et al., 2020).

Diagnosis dini untuk penyakit alzheimer sangat penting untuk menerapkan rencana perawatan yang tepat dan menunda perkembangan penyakit (Rasmussen & Langerman, 2019). Diagnosis yang efisien dan akurat juga penting untuk menghemat waktu dan mengurangi kesalahan. Selain itu adanya overlap yang dianggap dokter sebagai perubahan abnormal dan perubahan normal terkait usia (Maddury & Desai, 2022). Untuk mendeteksi Alzheimer para dokter juga melakukan pemindaian pada otak seperti *Computed Tomography* (CT), *Magnetic Resonance Imaging* (MRI), *Positron Emission Tomography* (PET). MRI merupakan pemeriksaan penting dalam kasus demensia. MRI aman karena tidak melibatkan radiasi *ionizing*, individu dapat dicitrakan secara serial tanpa khawatir tentang karsinogenesitas (Miglioretti et al., 2012). Namun penyakit alzheimer tidak terlalu terlihat jelas dalam gambar MRI otak seperti pada kasus dengan tumor, misalnya. Oleh karena itu, ini merupakan tantangan serius dalam ekstraksi fitur gambar (Awwal, 2021).

EfficientNet pertama kali diusulkan oleh (Tan & Le, 2019) Arsitektur *EfficientNet* menggunakan metode scaling dimana menggunakan komposit sederhana dan efisien untuk mengubah skala *depth*, *width*, dan *resolution* secara seimbang, sehingga meningkatkan akurasi model secara signifikan (P. Zhang et al., 2020). *EfficientNetB3* salah satu model *EfficientNet* yang menawarkan keseimbangan antara akurasi dan sumber daya komputasi berbanding lurus dibanding varian lainnya (Batool & Byun, 2023), kekurangan *EfficientNetB3* menggunakan aturan penskalaan majemuk dimana secara agresif meningkatkan ukuran gambar yang menyebabkan konsumsi memori besar dan pelatihan yang lambat, serta menggunakan *MBConv* layer dimana *depthwise convolutions* memiliki parameter dan *FLOPs* yang lebih sedikit dari *regular convolutions* sehingga biasanya tidak dapat memanfaatkan sepenuhnya *modern accelerators*. Kekurangan tersebut diperbaiki pada *EfficientNetV2* yaitu model yang dirancang berdasarkan *progressive learning*, menaikkan kecepatan proses pelatihan dengan sedikit kenaikan parameter dan *FLOPs*. Lebih memanfaatkan *mobile* atau *server accelerators* dengan kombinasi *MBConv* dan *FusedMBConv*, perbedaannya *depthwise conv3x3* dan *Conv1x1* pada *MBConv* diganti dengan *single Conv3x3* pada *FusedMBConv*. *EfficientNetV2S* merupakan varian *EfficientNetV2* yang lebih kecil dibandingkan dengan versi *L* atau *M* ukuran model dan jumlah parameternya (Tan & Le, 2021). Walaupun kecepatan pelatihan semakin ditingkatkan dengan *progressive learning* namun sering mengakibatkan penurunan akurasi dan rentan terhadap *overfitting*, yang dapat diatasi dengan regularisasi adaptif seperti *dropout* dan *data augmentation* (Tummala et al., 2023).

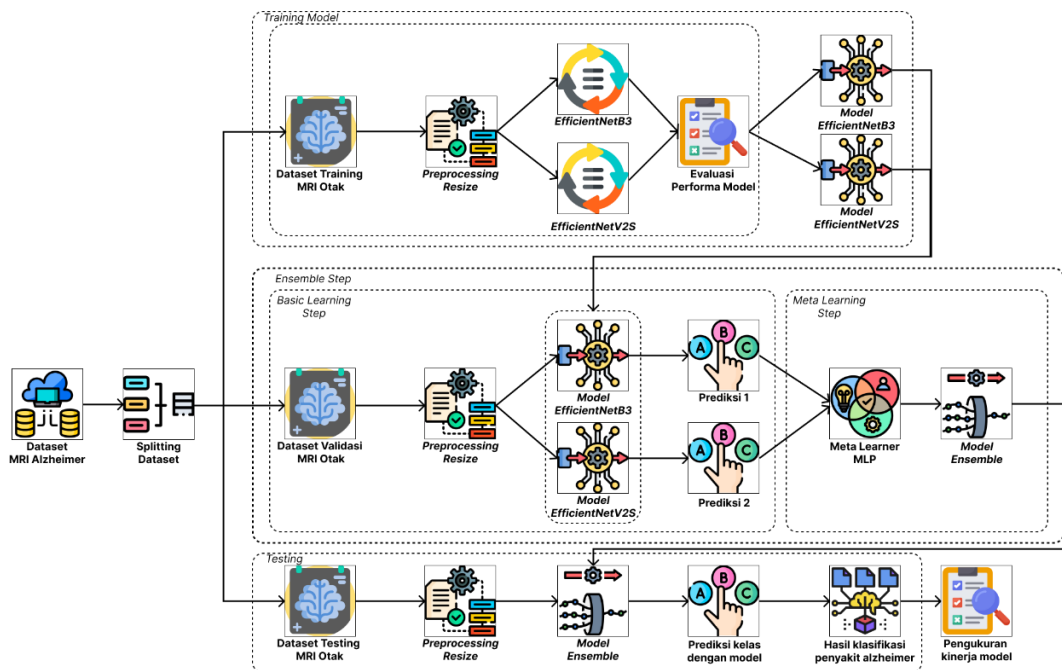
Penelitian yang dilakukan menjadi rujukan pada penelitian ini yaitu penelitian yang telah dilakukan sebelumnya oleh (Awwal, 2021) dalam melatih model dari 5 arsitektur CNN dengan metode *transfer learning*, yaitu *VGG19*, *Inception-ResNet-v2*, *ResNet152v2*, *EfficientNetB5* dan *EfficientNetB6* serta melakukan *ensembling* model dengan *average ensembling* pada beberapa kombinasi. Hasil penelitian tersebut *EfficientNetB5* memiliki 92,2%, *EfficientNetB6* memiliki 87,1%, dan *ensemble* kedua model 92% validasi akurasi, namun pada penelitian tersebut dataset yang digunakan lebih sedikit yaitu 416 data seluruh kelas dan perlu

menggunakan *pre-trained models*. Hal tersebut menjadi dasar pada penelitian ini untuk menggunakan dataset yang lebih banyak sehingga tanpa perlu menggunakan *pre-trained models* dengan kombinasi *output* model *EfficientNetB3* dan *EfficientNetV2S* untuk meningkatkan akurasi.

Berdasarkan penelitian sebelumnya, pada penelitian ini melakukan eksperimen dengan melakukan evaluasi performa pada arsitektur *EfficientNetB3* dan *EfficientNetV2S* serta *ensemble model* untuk mendapatkan model yang efektif dan efisien dengan menggunakan data augmentasi pada studi kasus klasifikasi jenis penyakit alzheimer.

2. METODE PENELITIAN

Pada penelitian ini, proses pelatihan dilakukan masing – masing arsitektur *EfficientNetB3* dan *EfficientNetV2S*. Dalam prosesnya pengaturan *hyperparameter* dilakukan untuk mendapatkan *output model* terbaik. Selain itu dilakukan proses *ensemble stacking* untuk meningkatkan *accuracy* dari *base model* (*EfficientNetB3* dan *EfficientNetV2S*). Berikut adalah proses yang dilakukan dari awal penelitian sampai hasil pengujian.



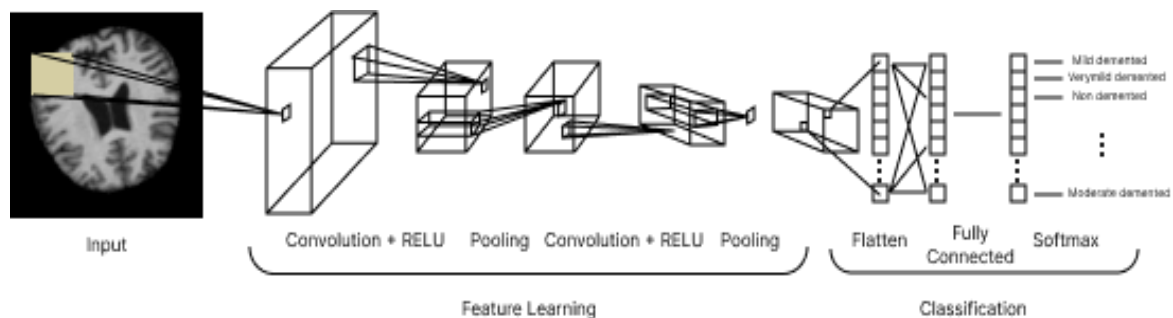
Gambar 1 Alur Penelitian

Pada Gambar 1 menunjukkan proses sistem diawali dengan pembagian *dataset* menjadi tiga, yaitu *training*, *validation*, dan *testing*. Data *training* merupakan citra yang akan digunakan pada proses pelatihan dengan melalui proses *preprocessing resize*, kemudian citra dilakukan proses pelatihan pada *EfficientNetB3* dan *EfficientNetV2S* dengan pengaturan *hyperparameter* seperti *learning rate*, *epoch* dan penambahan layer yang sudah ditentukan. *Output* dari proses ini menghasilkan *model* dari kedua arsitektur. Selanjutnya dilakukan proses *ensemble stacking* yang terbagi menjadi tahap *basic learning* dan *meta learning*. Pada tahap *basic learning* menggunakan *output model* dari kedua arsitektur sebagai *input*, pada proses ini akan dilakukan prediksi menggunakan *model EfficientNetB3* dan *EfficientNetV2S* terhadap seluruh data *validation*. Hasil prediksi kedua model digunakan sebagai *input* pada tahap *meta learning*

yaitu melakukan pelatihan baru dengan MLP (*Multilayer Perceptron*) yang akan menghasilkan *output* model *ensemble*. Tahap selanjutnya dilakukan proses pengujian dengan menggunakan data *testing* dari model yang telah dibuat dengan *accuracy*, *precision*, *recall*, dan *f1-score*. Setelah pengujian dilakukan selanjutnya evaluasi model menggunakan *confusion matrix* lalu memunculkan hasil yang didapatkan untuk mengetahui performa model yang dihasilkan.

2.1. Deep Learning

Deep learning menggabungkan kemajuan dalam daya komputasi dan jaringan saraf bekerja dengan banyak lapisan untuk mempelajari pola rumit dalam sejumlah besar data. **(Muniasamy & Alasiry, 2020)**. *Convolutional Neural Network* (CNN) merupakan salah satu metode pada *deep learning*. CNN merupakan suatu sistem operasi konvolusi yang terdiri dari beberapa lapisan pemrosesan dengan memanfaatkan beberapa elemen dan bekerja secara paralel **(Santoso, 2018)**. CNN secara umum terletak pada *convolution* dan *pooling layer*. *Convolution* berfungsi untuk mengumpulkan aspek paling informatif sehingga dapat mengurangi besarnya ukuran dimensi pada data dan mengurangi kompleksnya perhitungan. Sedangkan *pooling layer* berfungsi untuk melakukan ekstraksi untuk mendapatkan informasi yang paling informatif setelah dilakukannya proses *convolution* **(Purba, 2020)**. Ilustrasi arsitektur CNN ditunjukkan pada Gambar 2



Gambar 2. Ilustrasi Arsitektur CNN

2.2. Adam Optimizer

Optimization atau optimasi adalah kunci bagaimana *neural network* dapat belajar. Hal ini mengacu pada proses untuk meminimalkan *error*, *cost*, dan juga menemukan lokasi *error* paling sedikit **(FIRNANDO, 2021)**. *Tensorflow* menyediakan beberapa opsi untuk pengoptimalan model seperti *Gradient Descent Optimizer*, *Adam Optimizer*, *RMSProp Optimizer*, *Adagrad Optimizer*, *Ftrl Optimizer* dan *Momentum Optimizer* **(Mukhopadhyay, 2018)**. Adam atau *adaptive moment estimation* merupakan salah satu algoritma optimisasi yang banyak digunakan dalam deep learning. Adam menggabungkan kelebihan dari *AdaGrad Optimizer* dan *RMSProp Optimizer* yaitu dapat menangani masalah *sparse gradients* dan juga bekerja dengan baik dan memanfaatkan rata – rata pada momen pertama dan momen kedua dari gradien untuk mengadaptasi parameter *learning rate*, yang berbeda dengan *RMSProp* yang hanya memanfaatkan momen sementara **(Soydaner, 2020)**. Ada beberapa ketentuan yang penting dari algoritma Adam. Yaitu momentum tergabung langsung sebagai perkiraan momen orde pertama dari gradien. Kemudian Adam juga mencakup koreksi bias untuk perkiraan momen orde pertama dan momen orde kedua untuk memperhitungkan inisiasi awal pada titik nol. Algoritma Adam melakukan pembaruan rata – rata pergerakan eksponensial dari gradien m_t dan gradien kuadrat u_t , dimana *hyperparameter* ρ_1 dan ρ_2 mengendalikan nilai penurunan eksponensial dari rata – rata pergerakan. Rata – rata pergerakan ini merupakan perkiraan dari momen pertama (rata – rata) dan momen kedua (variansi tak terpusat) dari gradien **(Soydaner, 2020)**.

Algoritma Adam memerlukan momen pertama dan momen kedua dengan variabel m dan u . Setelah melakukan perhitungan gradien, perkiraan momen pertama dan momen kedua yang mengalami bias akan diperbarui setiap waktu langkah t masing – masing seperti yang ditunjukkan pada Persamaan (1) dan Persamaan (2) (**Soydaner, 2020**).

$$m_t \leftarrow \rho_1 m_{t-1} + (1 - \rho_1) g_t \quad (1)$$

$$u_t \leftarrow \rho_2 u_{t-1} + (1 - \rho_2) g \odot g \quad (2)$$

Setelah bias dikoreksi pada momen pertama dan momen kedua, kemudian dengan menggunakan momen yang sudah diperbaiki untuk mengestimasi pembaruan parameter yang akan dihitung dan diterapkan yang ditunjukkan pada Persamaan (3) hingga Persamaan (6) (**Soydaner, 2020**).

$$\hat{m}_t \leftarrow \frac{m_t}{1 - \rho_1^t} \quad (3)$$

$$\hat{u}_t \leftarrow \frac{u_t}{1 - \rho_2^t} \quad (4)$$

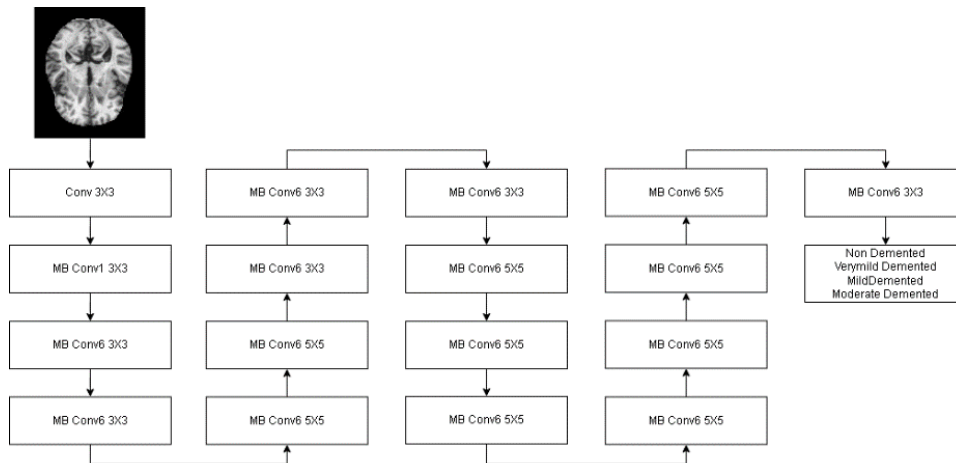
$$\Delta\theta = -\epsilon \frac{\hat{m}_t}{\sqrt{\hat{u}_t} + \delta} \quad (5)$$

$$\theta_t \leftarrow \theta_{t-1} + \Delta\theta \quad (6)$$

Adam memiliki beberapa kelebihan. Diantaranya hanya membutuhkan sedikit *tuning* pada bagian *learning rate*. Selain itu, metode yang mudah diterapkan dan tidak terpengaruh oleh perubahan skala diagonal gradien. Adam efisien secara komputasi dan memiliki kebutuhan memori yang sedikit (**Soydaner, 2020**).

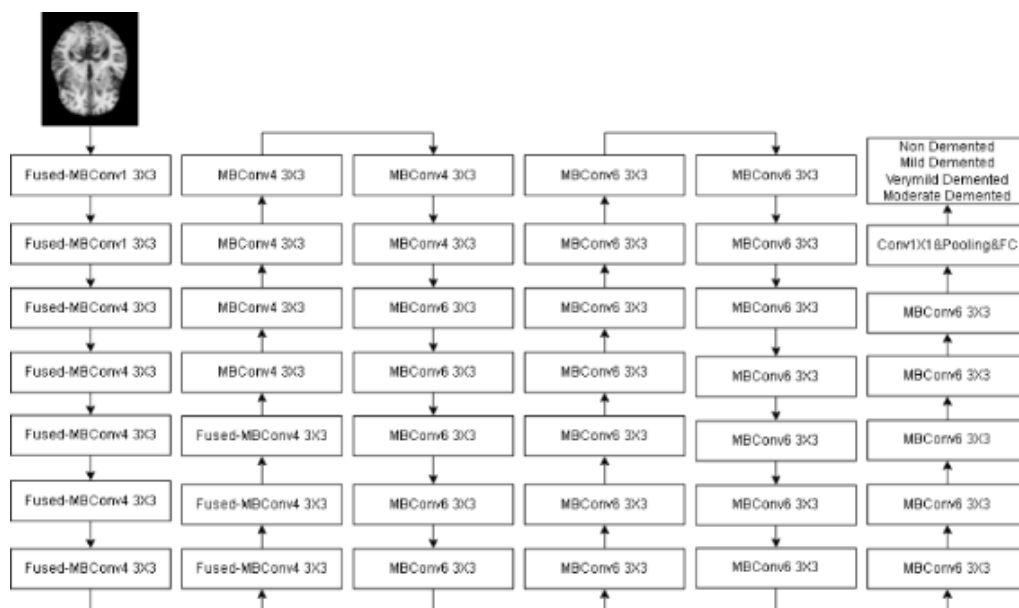
2.3. EfficientNetB3

Diantara beberapa varian *EfficientNetB0 – B7*, *EfficientNetB3* merupakan kombinasi yang seimbang antara sumber daya pemrosesan dan akurasi, sementara varian lain lebih akurat tetapi membutuhkan sumber daya komputasi yang besar atau varian yang lebih efisien tetapi kurang akurat (**Batool & Byun, 2023**). Walaupun masih menggunakan *MBConv* layer dimana *depthwise convolutions* memiliki parameter dan *FLOPs* yang lebih sedikit sehingga biasanya tidak dapat memanfaatkan sepenuhnya *modern accelerators* (**Tan & Le, 2021**) serta masih menggunakan aturan penskalaan majemuk dimana secara agresif meningkatkan ukuran gambar yang menyebabkan konsumsi memori besar dan pelatihan yang lambat (**Batool & Byun, 2023**). Ilustrasi dari struktur arsitektur *EfficientNetB3* ditunjukkan pada Gambar 3.

Gambar 3 Arsitektur *EfficientNetB3*

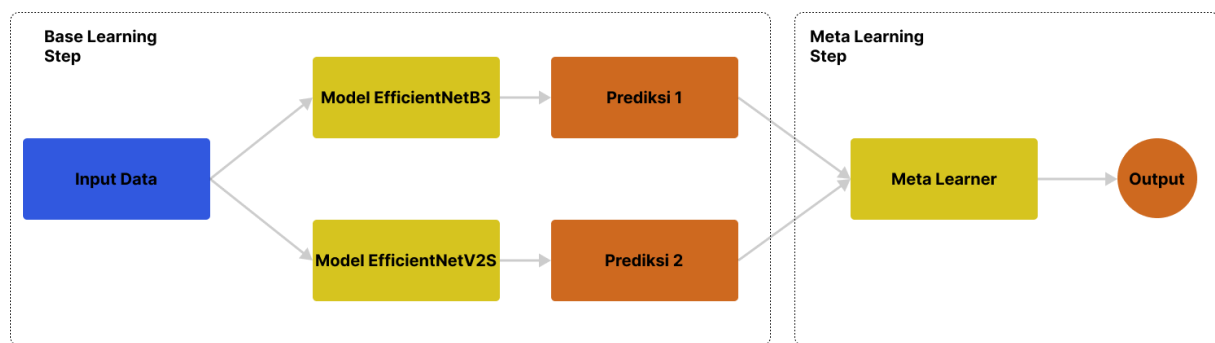
2.4. *EfficientNetV2S*

EfficientNetV2 adalah model yang dirancang berdasarkan *progressive learning* untuk mengatasi kekurangan *EfficientNet* untuk menaikkan kecepatan proses pelatihan dengan sedikit kenaikan parameter dan FLOPs (Tan & Le, 2021). Lebih memanfaatkan *mobile* atau *server accelerators* dengan mengganti *depthwise conv3x3* dan *Conv1x1* pada *MBConv* dengan *single Conv3x3*. *EfficientNetV2S* merupakan varian *EfficientNetV2* yang lebih kecil dibandingkan dengan versi *L* atau *M* ukuran model dan jumlah parameternya (Tan & Le, 2021). *EfficientNetV2S* menggunakan metode konvolusi baru yaitu dengan kombinasi seimbang antara *MBConv* yang ada pada *EfficientNet* dengan *FusedMBConv* yang menghasilkan kenaikan performa dengan model yang lebih kecil (Saeed et al., 2022). Meskipun 6,8 kali lebih kecil, *EfficientNetV2* melatih hingga 11 kali lebih cepat dibandingkan dengan *EfficientNetV1*. Walaupun kecepatan pelatihan semakin ditingkatkan dengan *progressive learning* namun sering mengakibatkan penurunan akurasi dan rentan terhadap *overfitting*, yang dapat diatasi dengan regularisasi adaptif seperti *dropout* dan *data augmentation* (Tummala et al., 2023). Ilustrasi dari struktur arsitektur *EfficientNetV2S* ditunjukkan pada Gambar 4.

Gambar 4 Arsitektur *EfficientNetV2S*

2.5. Ensemble Stacking

EfficientNetB3 merupakan kombinasi yang seimbang antara sumber daya pemrosesan dan akurasi, kekurangannya terdapat pada *MBConv* yang didalamnya terdapat *depthwise convolutions* yang memiliki lebih sedikit parameter dan *FLOPs* dari *regular convolutions*, sehingga biasanya tidak dapat memanfaatkan sepenuhnya *modern accelerators* seperti (TPU/GPU) (Tan & Le, 2021). Pada *EfficientNetV2S* menggunakan kombinasi dari *MBConv* dan *FusedMBConv* yang dapat mempercepat pelatihan dengan sedikit kenaikan pada parameter dan *FLOPs*, namun sering mengakibatkan penurunan akurasi dan rentan terhadap *overfitting* (Tummala et al., 2023). Untuk meningkatkan akurasi dan menurunkan tingkat *overfitting* digunakan teknik *ensemble stacking* dari *heterogeneous* model (T. Zhang & Li, 2021). *Stacking* generalisasi yaitu salah satu pendekatan *ensemble* dimana model baru belajar bagaimana mengintegrasikan prediksi terbaik dari beberapa model yang ada. Sehingga, model *ensemble CNN* yang tumpuk akan di-generalisasi dan sangat akurat (Gour & Jain, 2022). Teknik *stacking* bekerja dengan cara menggabungkan prediksi beberapa model yang disebut *weak-learner*, yang kemudian digabungkan menggunakan model yang disebut *meta learner*. *Stacking* terdiri dari dua level yaitu *base learner* sebagai level-0 dan *stacking model learner* sebagai level-1 atau *meta learner* (Wahid et al., 2022). Tahapan *ensemble stacking* dapat dilihat pada Gambar 5



Gambar 5 Tahapan *Ensemble Stacking*

2.6 Metrik Evaluasi

Evaluasi kinerja yang digunakan berupa *confussion matrix*, *precision*, *recall*, *accuracy*, dan *F1-score* sebagai pengujian pada performa model. Pada *confussion matrix* terdapat istilah – istilah yang digunakan yaitu:

TP (*True Positive*) = jumlah data positif yang diprediksi benar

TN (*True Negative*) = jumlah data negatif yang diprediksi benar

FN (*False Negative*) = jumlah data negatif yang diprediksi salah

FP (*False Positive*) = jumlah data positif yang diprediksi salah

Accuracy : Rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Biasanya akurasi suatu model ditentukan dalam bentuk persen. Perhitungan *accuracy* pada persamaan (7)

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (7)$$

Precision : Rasio prediksi benar dibandingkan dengan keseluruhan hasil yang terprediksi positif. Perhitungan *precision* pada persamaan (8)

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

Recall : Rasio prediksi benar dibandingkan dengan keseluruhan data yang terprediksi benar atau menghitung kekeliruan klasifikasi *false negative* pada model. Perhitungan *recall* pada persamaan (9)

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

F1-Score : Perbandingan rata – rata antara nilai *precision* dan nilai *recall*. Perhitungan *f1 score* pada persamaan (10)

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

3. HASIL DAN PEMBAHASAN

3.1 Penerapan *Datasets*

Datasets yang digunakan adalah *Augmented Alzheimer MRI Datasets* yang diambil dari Kaggle, dibuat oleh Uraninjo dengan menggunakan data orisinil yang dibuat oleh Sarvesh Dubey dari berbagai website. Augmentasi dilakukan menggunakan teknik *Resize*, *ShiftScaleRotate*, *RGBShift*, *Random BrightnessContrast*, *ColorJitter*, *Normalize*, dan *ToTensorV2*. Pada *dataset* yang digunakan dibagi menjadi data *training*, *validation*, dan *testing*. Pembagian tersebut meliputi 80% data *training*, 20% data *validation* yang diambil dari data augmentasi yang dipilih secara acak, sementara untuk data *testing* yang digunakan merupakan data orisinil. Rincian menggunakan *dataset* pada penelitian dapat dilihat pada Tabel 1.

Tabel 1 Rincian Penggunaan Dataset

No	Class	Data			
		Training	Validation	Testing(Orisinil)	Total Keseluruhan
1	<i>Non Demented</i>	9600	1920	3200	14720
2	<i>VeryMild Demented</i>	8960	1792	2240	12992
3	<i>Mild Demented</i>	8960	1792	896	11648
4	<i>Moderate Demented</i>	6464	1293	64	7821
Total Data		27187	6797	6400	40384

3.2 Pelatihan Model (*Model Training*)

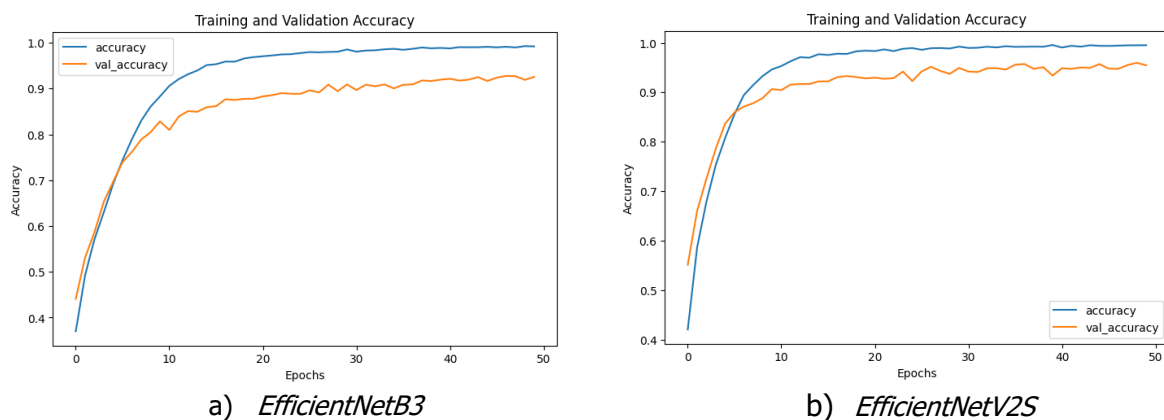
Pada penelitian yang dilakukan dengan membuat model dengan pengaturan *hyperparameter* yang telah ditentukan (Tabel 2) berdasarkan arsitektur *EfficientNetB3* dan *EfficientNetV2S*.

Tabel 2 *Hyperparameter* Setiap Model

Model	<i>Hyperparameter</i>
<i>EfficientNetB3</i>	<i>Adam Optimizer</i>
	<i>Learning Rate 0,0001</i>
	<i>Batch Size 32</i>
<i>EfficientNetV2S</i>	<i>Adam Optimizer</i>
	<i>Learning Rate 0,0001</i>
	<i>Batch Size 32</i>

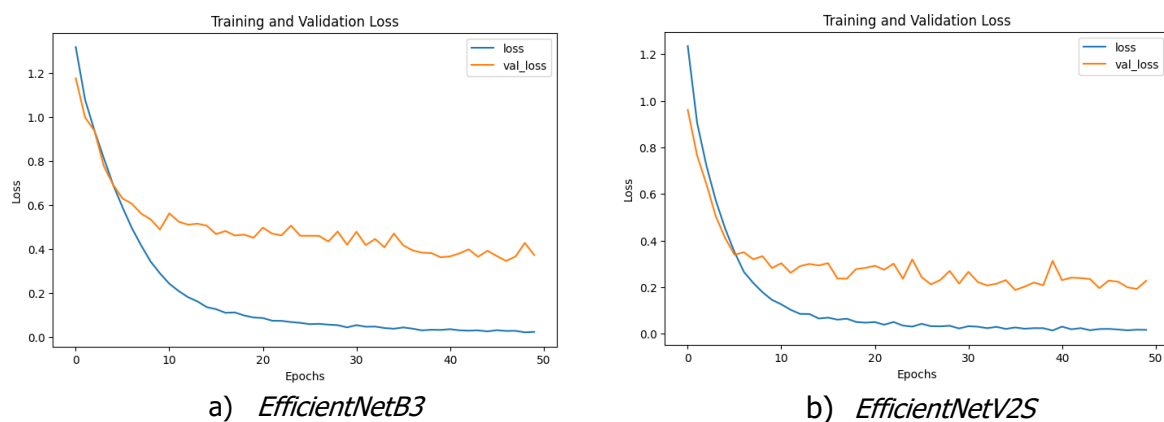
*Evaluasi Ensemble Stacking Arsitektur EfficientNetB3 dan EfficientNetV2S
(Studi Kasus Klasifikasi Penyakit Alzheimer)*

Pada tahap pelatihan menggunakan beberapa *hyperparameter* yang sudah tercantum pada Tabel 2, kemudian untuk setiap model menggunakan *epoch* yang sama yaitu 50 *epoch*. Kedua model tersebut akan dilakukan pelatihan dengan melihat hasil *accuracy* dan *validation accuracy*, dengan menggunakan *dataset* yang telah dibagi yaitu data *training* dan data *validation*. Model dapat dikatakan baik ditandai dengan *accuracy validation* yang mendekati hasil *accuracy* sehingga model dapat menggeneralisasi pada data baru. Dari hasil percobaan tersebut didapatkan grafik hasil *accuracy* dan *accuracy validation* ditunjukkan pada Gambar 6 (a dan b). Berdasarkan grafik hasil pelatihan lebih baik pada *EfficientNetV2S* dibandingkan dengan *EfficientNetB3* karena jarak antara *accuracy* dengan *accuracy validation* yang lebih sedikit.



Gambar 6 Grafik Pelatihan Accuracy

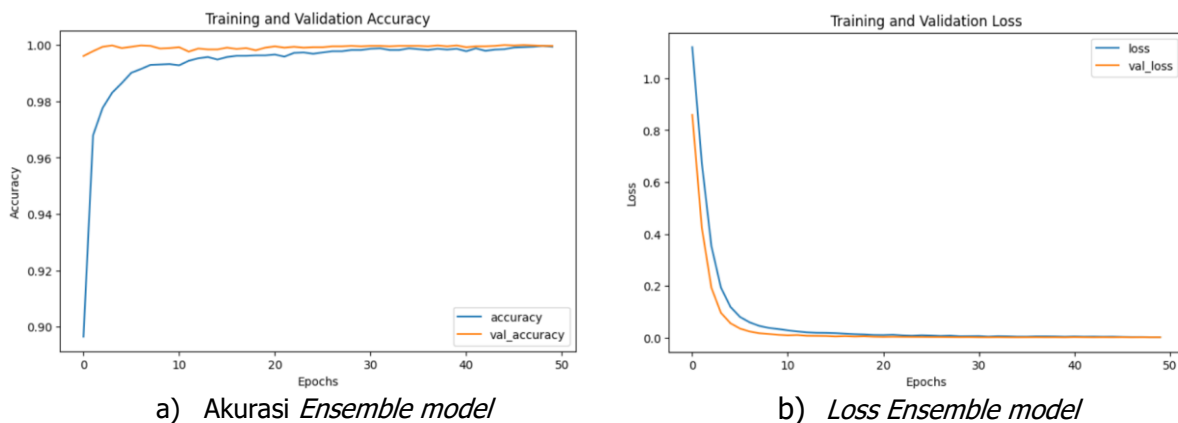
Selain *accuracy* dan *accuracy validation* terdapat *loss* dan *loss validation*. Selama pelatihan satu kali pengulangan (*epoch*) diakhir setiap *epoch* model dievaluasi menggunakan data *validation* dan *loss validation* dihitung menggunakan fungsi *loss* yang digunakan (Srinivasan et al., 2019). Nilai *loss* yang baik yaitu mendekati 0. Pada Gambar 7 (a dan b) menunjukkan hasil *loss* kedua model. Jika nilai *loss validation* mulai naik menandakan adanya *overfitting* pada model atau model terlalu bagus pada data pelatihan tetapi tidak dapat mengenali data baru. Berdasarkan grafik *EfficientNetB3* walaupun mengalami penurunan secara berkala setiap *epoch*, namun jarak antara *loss* dan *loss validation* pada *EfficientNetB3* lebih besar jika dibandingkan dengan *EfficientNetV2S*.



Gambar 7 Grafik Pelatihan Loss

Berdasarkan hasil pelatihan pada Gambar 6 dan Gambar 7, untuk meningkatkan *accuracy* dan menurunkan tingkat kesalahan (*loss*) pada model dilakukan pelatihan *ensemble stacking*

dengan menggunakan *model* yang didapat pada proses pelatihan sebagai *input*. Dalam prosesnya terbagi menjadi level 0 (*base learner*) dan level 1 (*meta learner*), pada level 0 *input data validation* dilakukan prediksi pada model *EfficientNetB3* dan *EfficientNetV2S* yang akan menghasilkan *output* berupa hasil prediksi masing – masing model terhadap *data validation* tersebut. Pada tahap level 1 hasil probabilitas prediksi tersebut digabungkan dengan fungsi *concatenate* yang digunakan sebagai *input* set pelatihan yang baru selanjutnya dilatih pada *meta model* dengan menggunakan MLP (*Multilayer Perceptron*) dengan 50 *epoch*. Gambar 8a menunjukkan grafik hasil *accuracy* dan Gambar8b menunjukkan grafik *loss* pada *ensemble model* yang telah dibuat. Berdasarkan grafik hasil tersebut selain adanya peningkatan pada *accuracy* menunjukkan model yang lebih stabil jika dibandingkan dengan model *EfficientNetB3* dan *EfficientNetV2S*.



Gambar 8 Grafik pelatihan *ensemble model*

Selain Grafik terdapat perbandingan hasil pelatihan pada model *EfficientNetB3*, *EfficientNetV2S* dan *ensemble model* terhadap data *validation* tercantum pada Tabel 3.

Tabel 3 Perbandingan Hasil Pelatihan

Model	Val Acc	Val Loss
<i>EfficientNetB3</i>	92,54%	0,3726
<i>EfficientNetV2S</i>	95,45%	0,2275
<i>Ensemble Model (B3+V2S)</i>	99,91%	0,0022

Berdasarkan hasil pelatihan pada Tabel 3, terlihat bahwa pada *EfficientNetB3* hasil *accuracy validation* lebih rendah yaitu 92,54% dengan *loss validation* 0,3726 dibandingkan dengan *EfficientNetV2S* 95,45% dengan *loss validation* 0,2275. Dari kekurangan dan kelebihan kedua model tersebut jika digabungkan menghasilkan kenaikan 99,91% *accuracy validation* dengan *loss validation* lebih rendah yaitu 0,0022.

3.3 Skenario Pengujian

Pengujian sistem dalam melakukan klasifikasi penyakit *alzheimer* dengan menggunakan arsitektur *EfficientNetB3*, *EfficientNetV2S*, dan *ensemble model*. Pengujian dilakukan dengan membandingkan beberapa model pelatihan, yang selanjutnya dilakukan pengujian menggunakan 6400 data *test*. Pengujian yang diukur yaitu *accuracy*, *precision*, *recall*, *f1-score*, dan *confusion matrix*.

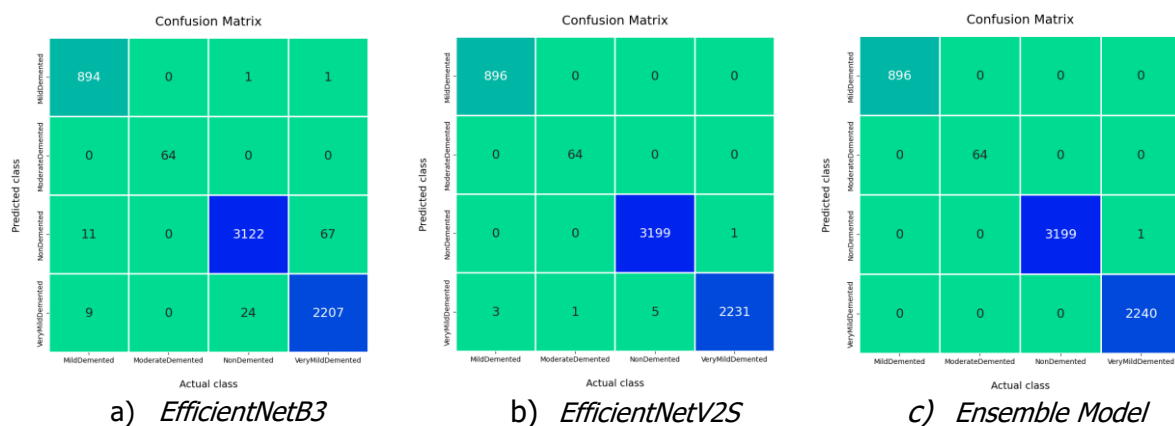
3.4 Hasil Pengujian Sistem

Setelah tahapan pelatihan selesai selanjutnya masuk ke tahapan pengujian model. Pada Tabel 4 menunjukkan pengujian dengan menghitung *accuracy*, *precision*, *recall*, dan *f1-score*.

Tabel 4 Hasil Pengujian

Model	Accuracy	Precision	recall	F1-score
<i>EfficientNetB3</i>	98,23%	98,25%	98,23%	98,24%
<i>EfficientNetV2S</i>	99,84%	99,84%	99,84%	99,84%
<i>Ensemble Model (B3-V2S)</i>	99,98%	99,98%	99,98%	99,98%

Di dalam Tabel 4 dapat dilihat bahwa dengan melakukan *ensemble* model berdasarkan prediksi *base model EfficientNetB3* dengan *EfficientNetV2S* dapat meningkatkan *accuracy*, *precision*, *recall*, dan *f1-score*. Walaupun peningkatan tersebut tidak berbeda jauh nilainya. Adapun pengaruh dari *ensemble model* terhadap hasil prediksi akhir yaitu pada pengujian dengan menggunakan *confusion matrix*. Pada Gambar menunjukkan jumlah kesalahan prediksi pada model *EfficientNetB3*, *EfficientNetV2S*, dan *Ensemble Model*.



Gambar 9 Confusion Matrix

Pada Gambar 9 terlihat bahwa jumlah kesalahan prediksi pada model *EfficientNetB3* yaitu 113 kesalahan. Sedangkan pada model *EfficientNetV2S* mendapatkan kesalahan prediksi 10 kesalahan. Sedangkan pada *ensemble model* mendapatkan kesalahan prediksi 1 kesalahan, dimana menunjukan penurunan bias dari model yang sebelumnya.

4. KESIMPULAN

Penelitian ini mengimplementasikan klasifikasi citra MRI otak penyakit Alzheimer dengan pendekatan ensemble stacking dari model *EfficientNetB3* dan *EfficientNetV2S*. Dalam eksperimen, hasil tertinggi ditemukan dengan melakukan freezing layer pada bottom layer model *EfficientNetB3* dan *EfficientNetV2S* serta pengaturan MLP dengan nilai dense 256, menghasilkan nilai rata-rata akurasi, presisi, recall, dan f1-score sebesar 99,98%. Penerapan metode ensemble stacking efektif mengurangi tingkat kesalahan prediksi, model ensemble stacking memiliki 1 kesalahan prediksi dibandingkan dengan 113 kesalahan prediksi pada model *EfficientNetB3* dan 10 kesalahan prediksi pada model *EfficientNetV2S*.

DAFTAR RUJUKAN

- Adi Nugroho, P., Fenriana, I., & Arijanto, R. (2020). IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA EKSPRESI MANUSIA. *JURNAL ALGOR*, 2(1). <https://jurnal.buddhidharma.ac.id/index.php/algor/index>
- Awwal, A. (2021). *Alzheimer's Disease Detection and Classification using Transfer Learning Techniques and Ensembling Operations on Convolutional Neural Networks*.
- Batool, A., & Byun, Y. C. (2023). Lightweight EfficientNetB3 Model Based on Depthwise Separable Convolutions for Enhancing Classification of Leukemia White Blood Cell Images. *IEEE Access*, 11, 37203–37215. <https://doi.org/10.1109/ACCESS.2023.3266511>
- FIRNANDO, A. (2021). *PENERAPAN DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DENGAN ARSITEKTUR EFFICIENTNET UNTUK KLASIFIKASI ALZHEIMER* [UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM]. <http://repository.uin-suska.ac.id/id/eprint/57861>
- Gour, M., & Jain, S. (2022). Automated COVID-19 detection from X-ray and CT images with stacked ensemble convolutional neural network. *Biocybernetics and Biomedical Engineering*, 42(1), 27–41. <https://doi.org/10.1016/j.bbe.2021.12.001>
- Maddury, S., & Desai, K. (2022). *Using Deep Learning to Determine Amyloid Deposition through PET and Clinical Data for Alzheimer's Prognosis*. <https://doi.org/10.1101/2022.10.04.22280712>
- Miglioretti, D. L., Johnson, E., Lee, C., Feigelson, H. S., Flynn, M., Greenlee, R. T., Kruger, R. L., Hornbrook, M. C., Roblin, D., Solberg, L. I., Vanneman, N., Ma, S., Weinmann, A. E., & Williams, T. (2012). ORIGINAL CONTRIBUTION Scan for Author Video Interview Use of Diagnostic Imaging Studies and Associated Radiation Exposure for Patients Enrolled in Large Integrated. In *Health Care Systems*. www.jama.com.
- Mukhopadhyay, S. (2018). Deep Learning and Neural Networks. In *Advanced Data Analytics Using Python* (pp. 99–119). Apress. https://doi.org/10.1007/978-1-4842-3450-1_5
- Muniasamy, A., & Alasiry, A. (2020). Deep learning: The impact on future eLearning. *International Journal of Emerging Technologies in Learning*, 15(1), 188–199. <https://doi.org/10.3991/IJET.V15I01.11435>
- Purba, J. S. (2020). *Inflamasi dalam Patologi Penyakit Alzheimer* (Vol. 33).
- Rasmussen, J., & Langerman, H. (2019). Alzheimer's Disease – Why We Need Early Diagnosis. *Degenerative Neurological and Neuromuscular Disease*, Volume 9, 123–130. <https://doi.org/10.2147/dnnd.s228939>
- Saeed, A., Shoukat, S., Shehzad, K., Ahmad, I., Eshmawi, A. A., Amin, A. H., & Tag-Eldin, E. (2022). A Deep Learning-Based Approach for the Diagnosis of Acute Lymphoblastic Leukemia. *Electronics (Switzerland)*, 11(19). <https://doi.org/10.3390/electronics11193168>
- Santoso, A. , & G. A. S. T. (2018). *IMPLEMENTASI DEEP LEARNING BERBASIS KERAS UNTUK PENGENALAN WAJAH*.
- Soydaner, D. (2020). A Comparison of Optimization Algorithms for Deep Learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13). <https://doi.org/10.1142/S0218001420520138>
- Srinivasan, P. A., Guastoni, L., Azizpour, H., Schlatter, P., & Vinuesa, R. (2019). Predictions of turbulent shear flows using deep neural networks. *Physical Review Fluids*, 4(5). <https://doi.org/10.1103/PhysRevFluids.4.054603>
- Tan, M., & Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*.
- Tan, M., & Le, Q. V. (2021). *EfficientNetV2: Smaller Models and Faster Training*. <https://github.com/google/>
- Tummala, S., Thadikemalla, V. S. G., Kadry, S., Sharaf, M., & Rauf, H. T. (2023). EfficientNetV2 Based Ensemble Model for Quality Estimation of Diabetic Retinopathy Images from DeepDRiD. *Diagnostics*, 13(4). <https://doi.org/10.3390/diagnostics13040622>

- Wahid, M. I., Lawi, A., Muh, D. A., & Siddik, A. (2022). *Perbandingan Kinerja Model Ensembled Transfer Learning Pada Klasifikasi Penyakit Daun Tomat*.
- Zhang, P., Yang, L., & Li, D. (2020). EfficientNet-B4-Ranger: A novel method for greenhouse cucumber disease recognition under natural complex environment. *Computers and Electronics in Agriculture*, 176. <https://doi.org/10.1016/j.compag.2020.105652>
- Zhang, T., & Li, J. (2021). Credit Risk Control Algorithm Based on Stacking Ensemble Learning. *Proceedings of 2021 IEEE International Conference on Power Electronics, Computer Applications, ICPECA 2021*, 668–670. <https://doi.org/10.1109/ICPECA51329.2021.9362514>