JURNAL ILMU KOMPUTER
Volume 2 Nomor 1, Juli, 2024
ISSN: 3031-125X (ONLINE)
Penerbit: Program Studi Teknik Informatika S-2 Universitas Pamulang

Analisis Klasifikasi Gambar Deteksi Merokok dengan Metode CNN yang Ditingkatkan Menggunakan Model *Fine Tuning* pada Arsitektur *MobileNetV3L, EfficientNetV2M,* dan *Vision Transformer*

Nuriyadin

Program Studi Magister Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Banten

Email: nuriyadin1922@gmail.com

ABSTRACT

Smoke detection faces challenges in detecting small and common events such as smoking, using deep learning techniques. Issues like these have resulted in unsatisfactory privacy and accuracy models. In the EfficientNetV2M model, the author first uses Data Augmentation to increase the amount and diversity of training data by carrying out transformations on existing data. A lower learning rate allows smoother parameter updates and can improve the final performance of the model, Fine Tune the EfficientNetV2M Layer and Finding the Ideal Learning Rate with LearningRateScheduler Callback. The improved performance in terms of accuracy and robustness shows that this method can be used in related fields and represents significant progress in the field of burn detection an accuracy rate of up to 97%. In the MobileNetV3L model the author obtained lower resource usage results, namely with an accuracy rate of 87%. In the Vision Transformer model the author uses a custom ViT (Vision Transformer) model for the feature extraction stage, then applies PCA for dimensional problems, and finally uses the XGBoost model for the classification stage and gets very satisfying results, namely with an accuracy level of 96 %. Future efforts will focus on improving this technology and finding ways to use it in broader contexts.

Keywords: detection, smoking, MobileNetV3L, EfficientNetV2M, Vision Transformer.

ABSTRAK

Deteksi merokok menghadapi tantangan dalam mendeteksi kejadian kecil dan umum seperti merokok, menggunakan teknik pembelajaran mendalam. Masalah seperti ini telah menghasilkan wawasan dan akurasi model yang tidak memuaskan. Pada model *EfficientNetV2M* ini penulis terlebih dahulu menggunakan *Data Augmentation* untuk meningkatkan jumlah dan keragaman data pelatihan dengan melakukan transformasi pada data yang sudah ada. *Learning rate* yang lebih rendah memungkinkan pembaruan parameter yang lebih halus dan dapat meningkatkan kinerja akhir model, *Fine Tune* terhadap *EfficientNetV2M Layer* dan *Finding the Ideal Learning Rate with LearningRateScheduler Callback*. Peningkatan kinerja dalam hal akurasi dan ketahanan menunjukkan bahwa metode ini dapat digunakan di bidang terkait dan mewakili kemajuan signifikan dalam bidang deteksi merokok dengan tingkat akurasi mencapai 97%. Pada model *MobileNetV3L* penulis mendapatkan hasil penggunaan sumber daya yang lebih rendah, dengan tingkat akurasi 87%. Pada model *Vision Transformer* penulis menggunakan model *custom ViT (Vision Transformer)* untuk tahap ekstraksi fitur, kemudian menerapkan *PCA* untuk masalah dimensi, dan terakhir menggunakan model *XGBoost* untuk tahapan klasifikasi dan didapatkan hasil yang sangat memuaskan yaitu dengan tingkat akurasi 96%. Upaya di masa depan akan fokus pada peningkatan teknologi ini dan menemukan cara untuk menggunakan dalam konteks yang lebih luas.

Kata kunci: deteksi, merokok, *MobileNetV3L, EfficientNetV2M, ViT.*

1. PENDAHULUAN

Merokok di tempat umum adalah masalah kesehatan masyarakat yang signifikan. Penggunaan teknologi untuk mendeteksi aktivitas merokok dapat membantu mengurangi paparan asap rokok dan meningkatkan kualitas udara. Dalam konteks ini, algoritma *machine learning* serta *deep learning* yang digunakan yaitu *MobileNetV3L, EfficientNetV2M* dan *Hybrid ViT XGBoost* memiliki potensi untuk digunakan dalam mendeteksi merokok.

Jaringan saraf tiruan telah merevolusi banyak bidang kecerdasan mesin dengan memberikan akurasi yang lebih besar dalam tugas pengenalan gambar yang kompleks. Namun peningkatan integritas sering kali harus dibayar mahal. Jaringan tercanggih saat ini memerlukan sumber daya komputasi yang melampaui kemampuan banyak aplikasi seluler dan tertanam [1].

Penelitian ini akan membandingkan algoritma deteksi merokok menggunakan algoritma *MobileNetV3L, EfficientNetV2M* dan *Hybrid ViT XGBoost* dengan penelitian sejenisnya untuk menentukan mana yang paling efektif dan efisien dalam mengklasifikasikan gambar yang mengandung aktivitas merokok. Wawasan yang diharapkan mencakup keakuratan deteksi yaitu mengukur sejauh mana setiap algoritma dapat secara akurat mendeteksi aktivitas merokok dari kumpulan data gambar, kecepatan pemrosesan yaitu mengukur waktu yang dibutuhkan masing-masing algoritma untuk memproses dan mengklasifikasikan gambar, serta kompleksitas model yaitu menilai kebutuhan komputasi dan memori dari masing-masing algoritma.

MobileNetV2 adalah arsitektur *convolutional neural network (CNN)* yang dirancang untuk perangkat mobile dengan efisiensi komputasi tinggi. Menggunakan teknik *depthwise separable convolution* untuk mengurangi jumlah parameter dan operasi komputasi [2]. Keunggulannya yaitu efisiensi tinggi dan dapat diimplementasikan pada perangkat dengan sumber daya terbatas, tetapi memiliki kelemahan yaitu mungkin kurang akurat dibandingkan dengan model yang lebih kompleks seperti *ViT* untuk tugas deteksi yang sangat spesifik dan detail.

Model *EfficientNet* terdiri dari 8 model dari B0 hingga B7, dengan setiap nomor model berikutnya mengacu pada varian dengan lebih banyak parameter dan akurasi lebih tinggi. Arsitektur *EfficientNet* menggunakan pembelajaran transfer untuk menghemat

waktu dan kekuatan komputasi [3]. Akibatnya, ini memberikan akurasi yang lebih tinggi nilai dibandingkan model pesaing yang dikenal. Hal ini disebabkan oleh penggunaan penskalaan yang cerdas pada kedalaman, lebar, dan resolusi. Para penulis telah menggunakan model B4, karena berisi 19 juta parameter, itu layak untuk pengaturan eksperimental kami, seperti yang disertakan dalam B5, B6, dan B7 masing-masing parameter 30M, 43M dan 66M. Selain itu, penulis telah menggunakan kumpulan data terpisah untuk memvalidasi usulan Model CNN menggunakan gambar yang tidak disertakan selama pengujian dan fase pelatihan. Model yang diusulkan telah dievaluasi menggunakan validasi silang bertingkat stratifikasi 10 kali lipat [4].

EfficientNet berfungsi sebagai model tulang punggung untuk klasifikasi. Keuntungannya adalah menghemat sejumlah besar parameter dan biaya komputasi sambil menunjukkan akurasi yang serupa dengan beberapa lainnya model *CNN* konvensional, seperti *ResNet50, DenseNet201*, dan *Inception-ResNet-v2*. Selain itu, itu memungkinkan kita untuk menggunakan pendekatan pembelajaran transfer dengan menyediakan model terlatih dari skala besar kumpulan data (misalnya, *ImageNet*) [5]. *VGG16* dan *Inception-ResNet-V2* digunakan sebagai model komparatif, keduanya representative Model berbasis *CNN* dan telah banyak diadopsi dalam tugas klasifikasi citra medis. Juga, mereka menyediakan model terlatih untuk pembelajaran transfer, begitu pula *EfficientNet*. Di antara model *EfficientNet*, model *EfficientNetB0* paling dasar diadopsi dalam penelitian ini karena dari kumpulan data kecil yang digunakan. Jumlah parameternya adalah 14.718.788 untuk *VGG16* dan 54.349.028 untuk *Inception-ResNet-V2*, masing-masing. Sebagai perbandingan, *EfficientNetB0* memiliki angka terendah, yaitu 4.059.815. Bentuk masukan semua model ditetapkan pada 224 × 224 untuk perbandingan model tulang punggung yang adil [6].

Implementasi resmi *MobileNetV2* tersedia sebagai bagian dari pustaka model *TensorFlow-Slim* di (kode sumber *MobileNetV2*. Tersedia dari *https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet)*. Modul ini dapat berhasil diimplementasikan menggunakan fungsi standar di lingkungan modern manapun, memungkinkan model kami menangani teknologi terkini dan banyak fungsi menggunakan parameter standar.

Selain itu, modul verifikasi ini sangat cocok untuk desain seluler karena secara signifikan mengurangi penggunaan peralatan yang diperlukan sekaligus menentukan peralatan mana yang tidak akan terlihat bagus di lingkungan yang luas. Hal ini memberikan cache yang lebih cepat dan dikontrol perangkat lunak, sehingga mengurangi kebutuhan memori pada sebagian besar perangkat dengan perangkat keras [7].

Arsitektur *Vision Transformer (ViT)* menjadi semakin populer dan banyak digunakan untuk mengatasi visi komputer aplikasi [8]. Fitur utama mereka adalah kemampuan untuk mengekstrak informasi global melalui mekanisme perhatian diri, yang kinerjanya lebih baik dari sebelumnya jaringan saraf konvolusional. Namun, penerapan dan kinerja *ViT* terus berkembang seiring dengan ukurannya dan jumlah orang yang dapat dilatih parameter, dan operasi. Selain itu, biaya komputasi dan memori perhatian diri meningkat secara kuadrat seiring dengan peningkatan gambar resolusi [9].

Vision Transformer (ViT) Architecture menggunakan mekanisme *transformer* yang awalnya diperkenalkan dalam pemrosesan bahasa alami untuk mengolah data gambar. *ViT* mengubah gambar menjadi *patch* kecil dan memperlakukan setiap *patch* seperti *token* dalam *NLP*, memungkinkan model untuk memahami konteks global dari gambar secara lebih baik [9].

XGBoost Gradient Boosting Tree (XGBoost) adalah algoritma *boosting* yang kuat dan efisien untuk masalah klasifikasi dan regresi. Algoritma ini menggabungkan beberapa pohon keputusan yang lemah untuk membentuk model yang kuat [10].

Algoritma *XGBoost* memiliki keunggulan yaitu efisiensi tinggi dalam memproses data dan kemampuan untuk menangani data yang tidak terstruktur, tetapi memiliki kelemahan yaitu urang efektif untuk data yang sangat besar dan kompleks seperti gambar dibandingkan dengan model *deep learning* [11]. Penelitian lainnya eksperimen ekstensif pada tiga *dataset* data *EEG benchmark* menunjukkan keunggulan algoritma *MViT* yang diusulkan dibandingkan metode prediksi kejang yang canggih, mencapai sensitivitas prediksi 90,28–91,15% untuk data *EEG* invasif [12].

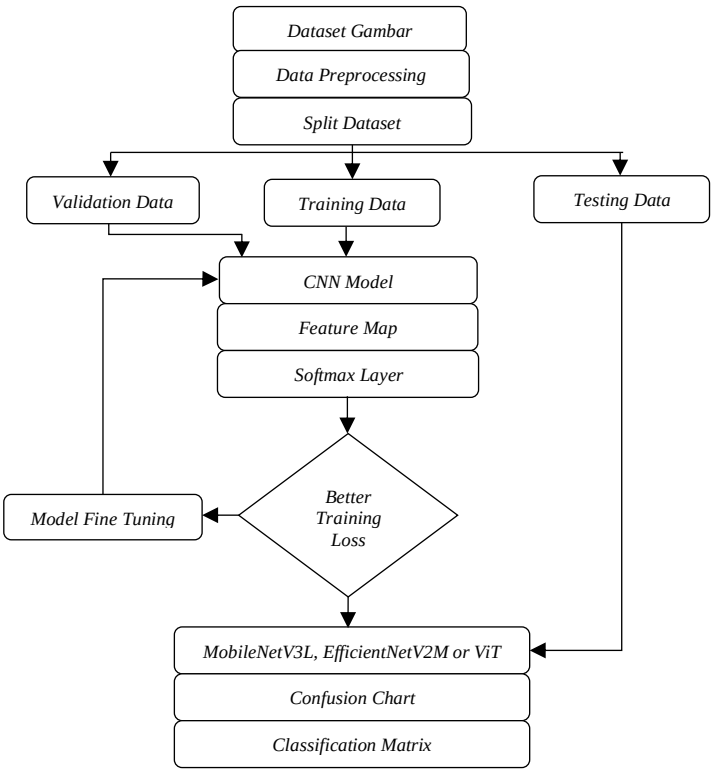
Penelitian sejenis berupa *Machine Learning and Vision Transformers for Thyroid Carcinoma Diagnosis* hasilnya menunjukkan peningkatan akurasi segmentasi secara signifikan dibandingkan dengan jaringan klasik, mencapai koefisien kesamaan *Dice*

sebesar 85,63% dan 81,64% serta nilai HD95 sebesar 14,53 dan 14,06 pada kumpulan data pribadi dan publik [13].

Penelitian ini memiliki rumusan tujuan penelitian mengevaluasi akurasi yaitu menentukan tingkat akurasi deteksi merokok dari algoritma *MobileNetV3L*, *EfficientNetV2M* dan *Hybrid ViT XGBoost*, menganalisis kecepatan pemrosesan yaitu membandingkan kecepatan pemrosesan gambar dibandingkan dengan algoritma lainnya yang sejenis, menilai efisiensi komputasi yaitu menilai kebutuhan komputasi (misalnya, penggunaan *CPU/GPU* dan memori) dari masing-masing algoritma, serta menentukan algoritma terbaik yaitu mengidentifikasi algoritma yang paling sesuai untuk digunakan dalam aplikasi deteksi merokok berdasarkan kriteria akurasi, kecepatan, dan efisiensi.

2. METODE

Penelitian deteksi merokok pada klasifikasi algoritma *MobileNetV3L*, *EfficientNetV2M* dan *Hybrid ViT XGBoost Architecture* menggunakan kumpulan data dengan jumlah 1.120 data yang berasal dari *Kaggle* dibagi menjadi dua kelompok. Setengah dari gambar tersebut masuk dalam kategori merokok dan sisanya masuk dalam kategori tidak merokok *(non-smoking).*



Gambar 1. Diagram Metode Penelitian

Dataset tersebut meliputi air minum, merokok, batuk, pernapasan, orang yang menelepon, orang yang merokok, orang, dan lainnya. Ini dirancang dengan memindai berbagai mesin pencari dengan memasukkan kata kunci seperti. Ini menghilangkan kebingungan antar kategori dan dengan demikian meningkatkan aplikasi pemodelan. Misalnya, kategori rokok berisi gambar berbagai jenis pembawa pesan dan simbol.

Selain itu, gambar pada kategori Tidak Merokok menunjukkan orang sedang merokok, minum air putih, merokok, memegang ponsel, batuk, dan lain-lain. Ini terdiri dari gambar orang yang menunjukkan tanda-tanda merokok, seperti: kumpulan data dapat digunakan di masa depan para peneliti akan mengembangkan algoritma untuk belajar mendeteksi dan mengevaluasi perokok untuk pemantauan lingkungan dan pemantauan kota pintar [14].

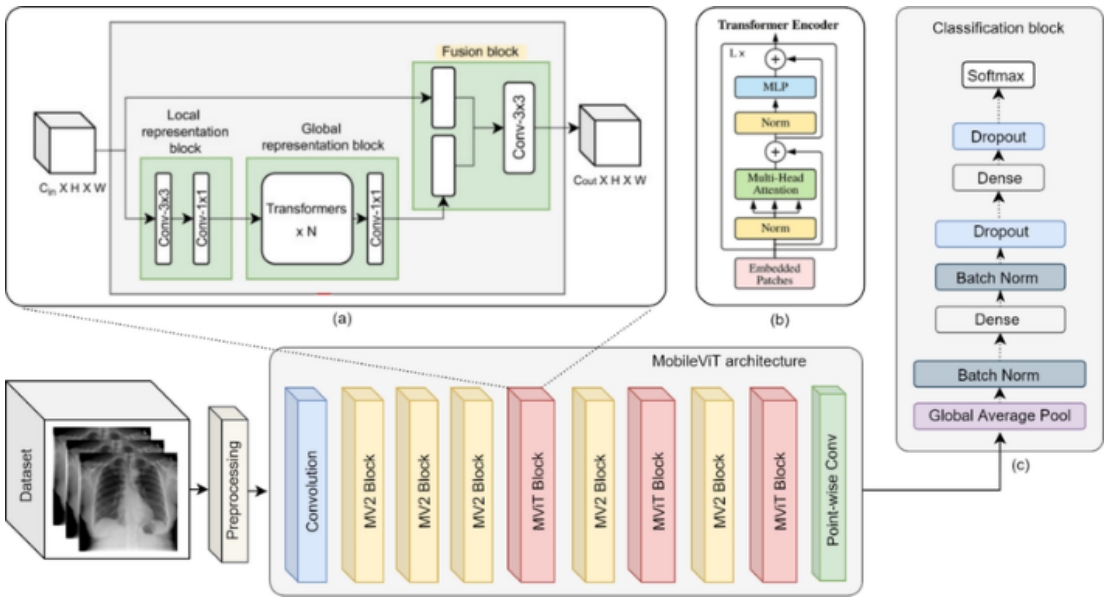
Tabel 1. Analisis Perbandingan Menggunakan Metode yang Berbeda

Method	Accuracy %	Params (mln)	Training Time (hr)
EfficientNetV2 [15]	85,27	55	13
ResNest [16]	86,61	48	4,27
MobileNetV3 [17]	82,14	1,77	1,52
ResNetD [18]	88,40	25	3,4
Levit [19]	94,64	18,9	8,6
Davit [20]	90,63	87,9	9,7
Coatlite [21]	91,10	45	4,7
ViT [22]	96,43	86	2,4
SmokerViT [23]	97,77	86,19	3,66
Ours			
MobileNetV3L	87,00	2,9	0,06
EfficientNetV2M	97,00	53,1	0,2
Hybrid ViT XGBoost	96,00	85,8	0,06

Pada Tabel 1, beberapa menggunakan arsitektur lainnya berdasarkan penelitian sebelumnya dengan menggunakan dataset yang sama dengan penulis lakukan, dan sebagian menggunakan dataset berbeda, sehingga bagian group yang didapatkan pada baris *Ours* merupakan arsitektur dalam penelitian naskah ini.

Pada arsitektur *MobileNetV3L* dimulai dengan tahapan *preprocessing* digunakan untuk inisialisasi data dan *list* label, melakukan *resize* gambar, mengkonversi ke format *array*, dan melakukan *scaling* pada *pixel intensities*. Selanjutnya melakukan *pre-processed image* dan label terkait pada data dan *list* label masing–masing, dan memastikan proses *training data* dalam format *array NumPy*. Semua gambar dalam kumpulan angka diproses dan diperkecil menjadi ukuran 224x224. Selanjutnya proses

pengujian pada model yang telah diusulkan sebelumnya untuk mendapatkan hasil kinerja. *Dataset* yang sudah disiapkan akan diuji dengan menggunakan *MobileNetV3L classifier* untuk *fine-tuning*. *Fine tuning* merupakan proses untuk *tweak* sebuah model *pre-trained* sehingga parameter akan beradaptasi dengan model baru [24].



Gambar 2. Arsitektur Model Algoritma *MobileNetV3L*

Pada model *EfficientNetV2M* dilakukan perubahan *hyperparameter* yang digunakan pada penelitian sejenisnya dan akan menggunakan *activation function ReLU* dan *GELU* sebagai pembanding, karena *GELU* dapat digunakan sebagai alternatif *ReLU* yang lebih kompleks [25].

Proses selanjutnya dilakukan proses *Data Augmentation* yaitu *random_flip_left_right, random_brightness, random_contrast, random_saturation,* serta *random_hue* untuk meningkatkan jumlah dan keragaman data pelatihan dengan melakukan transformasi pada data yang sudah ada. Augmentasi data membantu model untuk belajar lebih baik dan mengurangi *overfitting* dengan menghadirkan variasi data yang lebih luas.

Jika kumpulan data terlalu kecil, maka modelnya terlalu kecil berisiko mengalami *over-fitting*, yang berarti tidak dapat melakukan generalisasi secara efektif dan akan menghasilkan kinerja yang buruk pada kumpulan data baru. Oleh karena itu, untuk melatih kumpulan data kecil untuk pembelajaran mendalam, pekerjaan ini melakukan

data augmentasi untuk memiliki beberapa sampel pelatihan untuk mengatasi batasan ukuran kumpulan data. Penelitian ini menerapkan berbagai proses augmentasi pada dataset pelatihan, seperti yang diberikan pada tabel dibawah ini dengan melakukan berbagai augmentasi seperti mengubah ukuran, menskalakan, membalik, menggeser.

Tabel 2. Tabel Augmentasi dengan Algoritma *EfficientNetV2M*

Augmentation	***Value***
random_flip_left_right	*Left, Right*
random_brightness	*0,1*
random_contrast	*0,2 - 0,5*
random_saturation	*0,5 - 1*
random_hue	*0,2*

Create a Checkpoint Callback to Save the Model agar objek dapat diteruskan ke metode pelatihan model yang digunakan untuk melakukan tindakan tertentu pada titik-titik tertentu selama pelatihan. *Checkpoint callback* khususnya digunakan untuk menyimpan model pada interval tertentu atau saat kinerja tertentu tercapai. Create a Model Checkpoint Callback that Saves the Model's Weights Only dapat menyimpan bobot model tanpa menyimpan keseluruhan arsitektur model. Ini berguna untuk menghemat ruang penyimpanan ketika arsitektur model tidak berubah selama pelatihan.

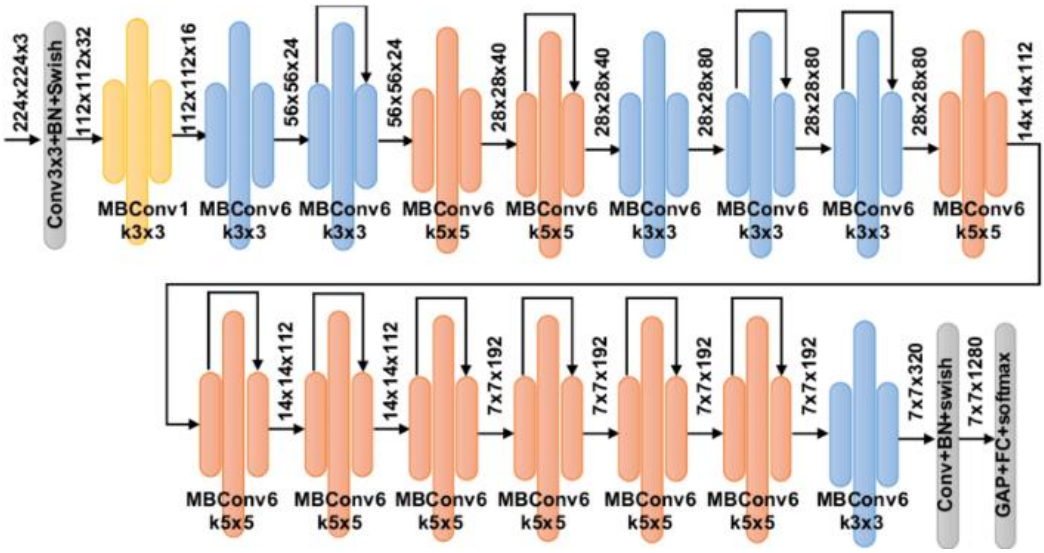
Proses *fine tuning* yang diterapkan pada metode ini yaitu *Create a New Instance of the Base Model with Lower Learning Rate* dengan membuat instance baru dari model dasar dengan *learning rate* yang lebih rendah dapat membantu dalam *fine*-tuning, terutama jika model sudah mendekati konvergensi. *Learning rate* yang lebih rendah memungkinkan pembaruan parameter yang lebih halus dan dapat meningkatkan kinerja akhir model.

Selanjutnya *Load the Weights from the Previous Model Checkpoint* memiliki peran bobot yang disimpan dari checkpoint sebelumnya dapat dimuat ke dalam model baru. Ini memastikan bahwa pelatihan dilanjutkan dari titik yang sama dengan model sebelumnya. Lalu melakukan evaluasi hasil pengetesan untuk memastikan bahwa hasilnya sama dilakukan setelah memuat bobot, model dievaluasi ulang pada data uji untuk memastikan bahwa kinerja model tetap konsisten dan tidak ada perubahan akibat pemuatan bobot. Proses *Fine Tune* pada *EfficientNetV2M Layer* melibatkan pelatihan ulang model dengan bobot awal yang berasal dari model yang sudah dilatih sebelumnya. Hanya beberapa lapisan terakhir yang biasanya disesuaikan, sementara lapisan awal tetap

beku. Perlu adanya proses pembekuan semua lapisan kecuali 10 lapisan terakhir berarti lapisan-lapisan tersebut tidak akan diperbarui selama pelatihan. Ini dilakukan untuk melindungi representasi fitur yang sudah dipelajari pada lapisan awal sambil menyempurnakan lapisan akhir untuk tugas spesifik.

Setelah membekukan lapisan, model harus dikompilasi ulang dengan *optimizer* dan *learning rate* yang sesuai untuk memulai pelatihan kembali, dan dilakukan pencarian *Ideal Learning Rate* dengan *LearningRateScheduler Callback* untuk menyesuaikan learning rate selama pelatihan berdasarkan epoch atau kondisi tertentu. Ini membantu menemukan learning rate yang optimal dengan memonitor kinerja model.

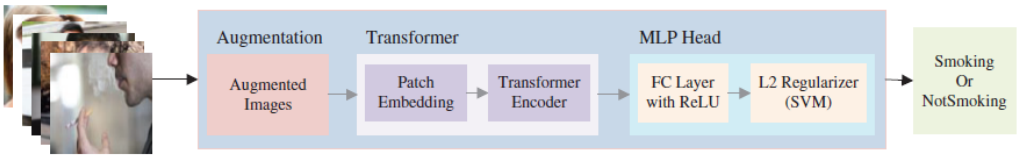
Setelah menggunakan LearningRateScheduler, learning rate yang memberikan kinerja terbaik dipilih berdasarkan grafik kinerja (seperti loss atau accuracy) untuk dilakukan evaluasi yang digunakan untuk memahami kinerja model klasifikasi. Ini menunjukkan jumlah prediksi benar dan salah yang dibuat oleh model untuk setiap kelas, memberikan gambaran tentang seberapa baik model mengklasifikasikan setiap kategori.



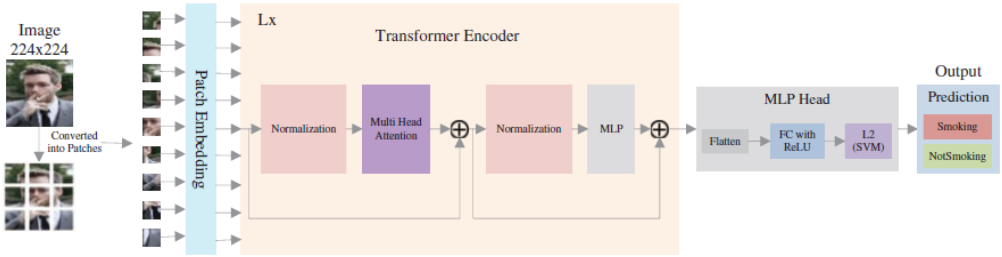
Gambar 3. Arsitektur Model Algoritma *EfficientNetV2M*

Pada model dan *Hybrid ViT XGBoost Architecture* mengambil rangkaian patch gambar sebagai masukan dan prediksi label kelas untuk gambar masukan. *ViT* berbeda dari CNN tradisional, yang memiliki fungsi komputasi menggunakan *array* piksel. *Transformer* membagi gambar menjadi potongan-potongan dengan ukuran tetap. Lalu memasukkan tambalan-tambalan ini ke dalam proyeksi linier dari lapisan penyematan tambalan rata untuk menghasilkan *vector* sering dikenal sebagai *token*. *Encoder*

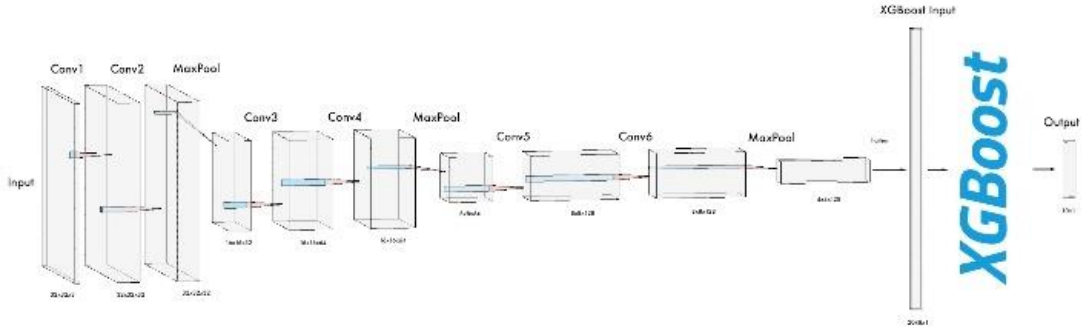
Transformer akan menerima *token* ini sebagai *patch* dan data lokasi. *Encoder Transformer* memiliki jumlah *output* yang sama dengan *input* [23]. Keluaran yang sesuai dengan kelas tersebut kemudian dimasukkan ke kepala *MLP* untuk mengeluarkan prediksi dan klasifikasi. Arsitektur *SmokerViT* diilustrasikan dalam gambar berikut.



Gambar 4. Mekanisme Kinerja dari *Vision Transformer*



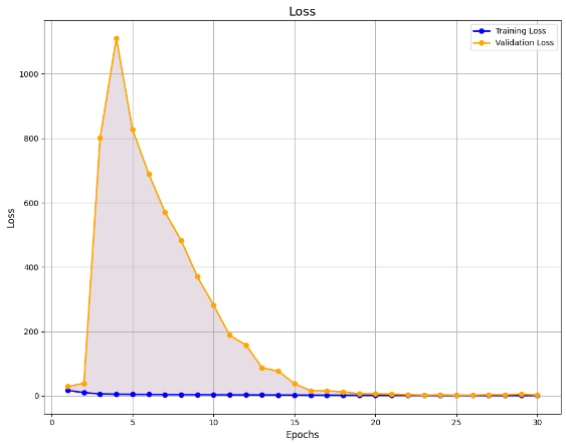
Gambar 5. Arsitektur yang Diusulkan dari *ViT*



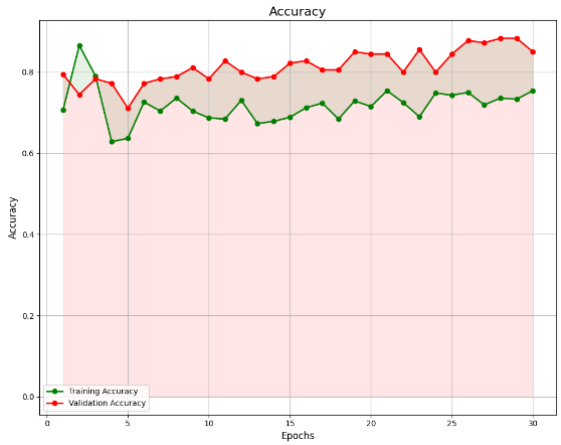
Gambar 6. Arsitektur yang Diusulkan dari *ViT* dengan *XGBoost*

3. HASIL DAN PEMBAHASAN

Dataset akan dikelompokan menjadi dua yaitu merokok dan tidak merokok, selanjutnya proses *training* dilakukan untuk menghasilkan *file* model yang nantinya di *generate* otomatis saat *training* selesai dan model bisa digunakan untuk proses klasifikasi, seperti yang dijelaskan. Pada arsitektur *MobileNetV3L* ada beberapa tahap yang dilalui seperti *initializing data, resizing image*, *conversion images to numpy array*, dan *scalling pixel intensities*. Dari hasil *training* akan didapatkan akurasi beserta grafik *plot* pada Gambar 7.



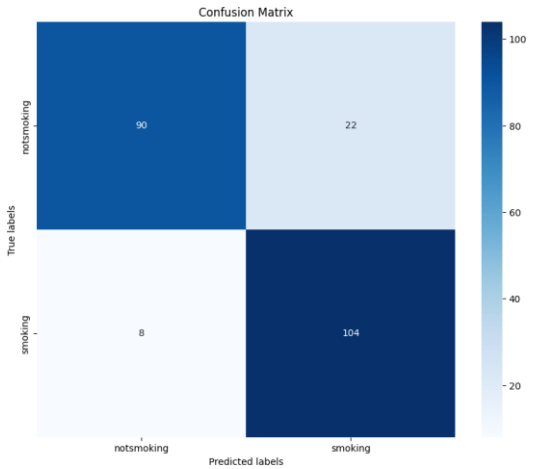
Gambar 7. Grafik *Plot* untuk *Training* dan *Validation Loss* pada Algoritma *MobileNetV3L*



Gambar 8. Grafik *Plot* untuk *Training* dan *Validation Accuracy* pada Algoritma *MobileNetV3L*

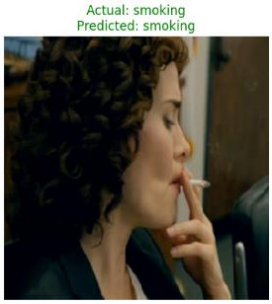
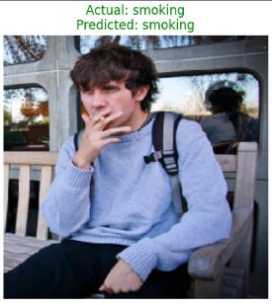
Tabel 3. Laporan Klasifikasi dengan Algoritma *MobileNetV3L*

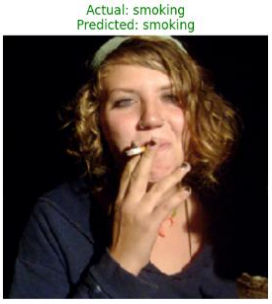
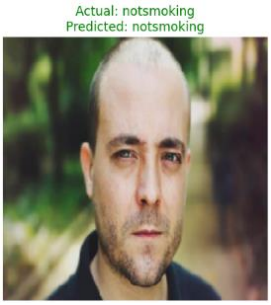
Classification Report	***Precision***	***Recall***	***F1-Score***	***Support***
Not smoking	*0,92*	*0,80*	*0,86*	*112*
Smoking	*0,83*	*0,93*	*0,87*	*112*
Accuracy			*0,87*	*224*
Macro avg	*0,87*	*0,87*	*0,87*	*224*
Weighted avg	*0,87*	*0,87*	*0,87*	*224*



Gambar 9. *Confusion Matrix* dengan Algoritma *MobileNetV3L*

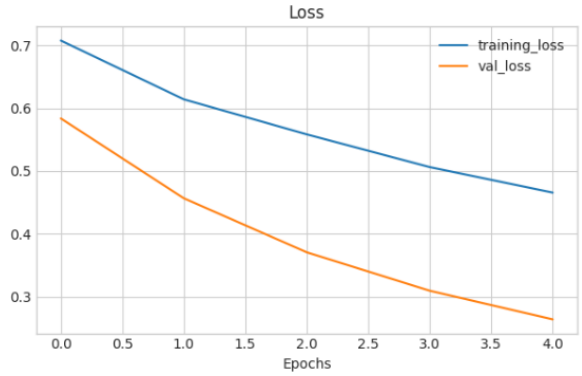
Tabel 4. Hasil Pengujian dengan Algoritma *MobileNetV3L*

No	Pengujian	Gambar
1	*Actual: notsmoking* *Predicted: smoking*	
2	*Actual: smoking* *Predicted: smoking*	
3	*Actual: smoking* *Predicted: smoking*	
4	*Actual: smoking* *Predicted: smoking*	
5	*Actual: smoking* *Predicted: smoking*	

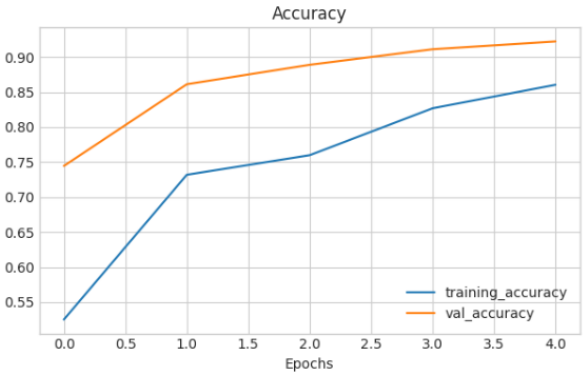
No	Pengujian	Gambar
6	*Actual: notsmoking* *Predicted: notsmoking*	
7	*Actual: smoking* *Predicted: smoking*	
8	*Actual: notsmoking* *Predicted: notsmoking*	
9	*Actual: smoking* *Predicted: smoking*	
10	*Actual: notsmoking* *Predicted: notsmoking*	

Pada arsitektur *EfficientNetV2M*, ada beberapa tahap yang dilalui seperti *Data Augmentation, Create a Checkpoint Callback to Save the Model, Create a*

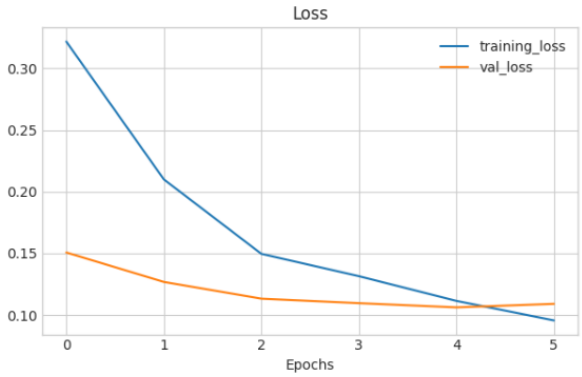
ModelCheckpoint Callback that Saves the Model's Weights Only, Compile and Fit, Test Results of the Model, Create a New Instance of the Base Model with Lower Learning Rate, Load the Weights from the Previous Model Checkpoint, Evaluate the Test Results to Make Sure They Are Same, Fine Tune the EfficientNetV2M Layer, Freeze All Layers Except for the Last 10, Recompiling the Model, Finding the Ideal Learning Rate with LearningRateScheduler Callback, Picking the Best Learning Rate from the Graph. Dari hasil *training* akan didapatkan *training loss* dan *validation loss*, serta *training accuracy* dan *validation accuracy* beserta masing-masing grafik plot.



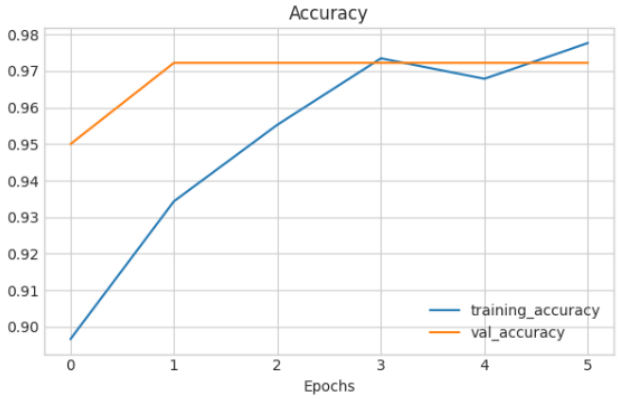
Gambar 10. Grafik *Plot* untuk *Training* dan *Validation Loss* pada Model 1 Algoritma *EfficientNetV2M*



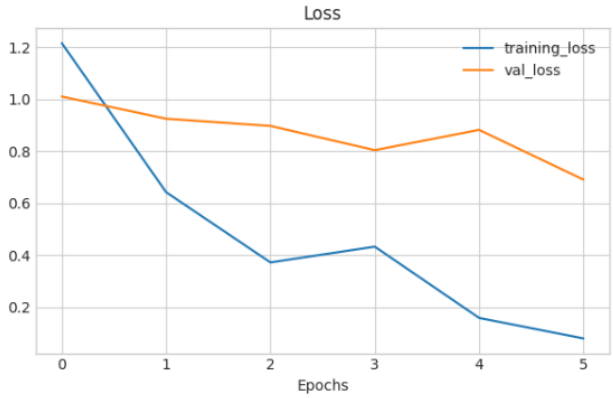
Gambar 11. Grafik *Plot* untuk *Training* dan *Validation Accuracy* pada Model 1 Algoritma *EfficientNetV2M*



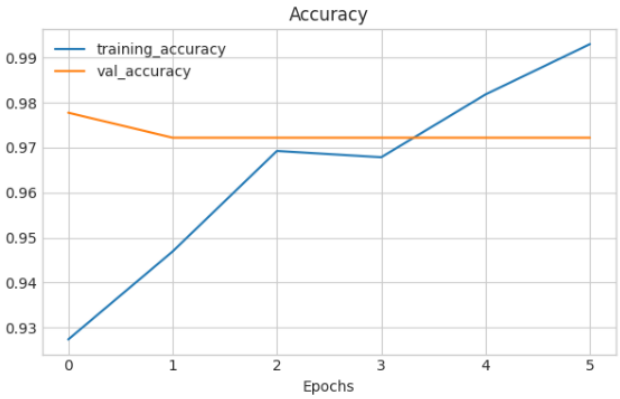
Gambar 12. Grafik *Plot* untuk *Training* dan *Validation Loss* pada Model 2 Algoritma *EfficientNetV2M*



Gambar 13. Grafik *Plot* untuk *Training* dan *Validation Accuracy* pada Model 2 Algoritma *EfficientNetV2M*



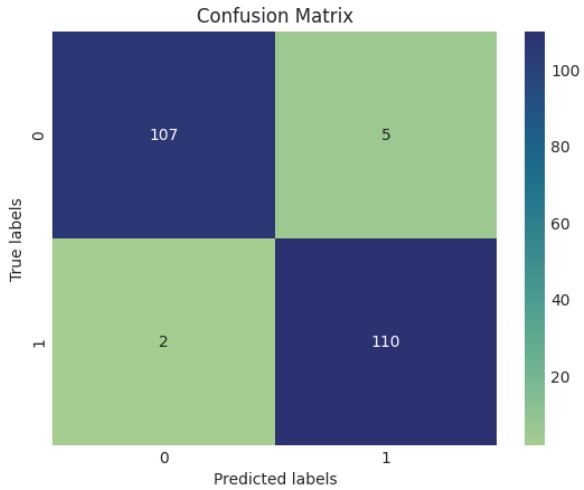
Gambar 14. Grafik *Plot* untuk *Training* dan *Validation Loss* pada Model 3 Algoritma *EfficientNetV2M*



Gambar 15. Grafik *Plot* untuk *Training* dan *Validation Accuracy* pada Model 3 Algoritma *EfficientNetV2M*

Tabel 5. Laporan Klasifikasi dengan Algoritma *EfficientNetV2M*

Classification Report	***Precision***	***Recall***	***F1-Score***	***Support***
Not smoking	*0,98*	*0,96*	*0,97*	*112*
Smoking	*0,96*	*0,98*	*0,97*	*112*
Accuracy			*0,97*	*224*
Macro avg	*0,97*	*0,97*	*0,97*	*224*
Weighted avg	*0,97*	*0,97*	*0,97*	*224*



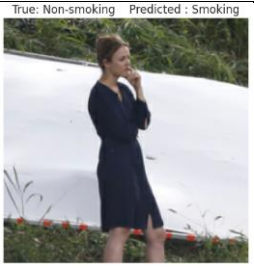
Gambar 16. *Confusion Matrix* dengan Algoritma *EfficientNetV2M*

Tabel 6. Hasil Pengujian Menggunakan Sumber Luar dengan Algoritma *EfficientNetV2M*

No	Sumber Pengujian	Hasil dan Gambar
1	*https://img.freepik.com/free-photo/young-man-smoking_144627-29295.jpg*	Smoking with probability 100.0 Non-Smoking with probability 5.9117654876278695e-18 
2	*https://encrypted-tbn0.gstatic.com/images?q=tbn:And9GcQZ2FnaBLHoNCw4OM00db5ahJdvs_LXEo45OQ&usqp=CAU*	Smoking with probability 100.0 Non-Smoking with probability 6.787019308277756e-14 
3	*https://t4.ftcdn.net/jpg/02/24/86/95/360_F_224869519_aRaeLneqALfPNBzg0xxMZXghtvBXkfIA.jpg*	Smoking with probability 1.1788005061007592e-13 Non-Smoking with probability 100.0 

No	Sumber Pengujian	Hasil dan Gambar
4	*https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcRY7f8d0V9xyQ37QTSgdDhFm6eZ15zdzoxYxw&usqp=CAU*	Smoking with probability 0.5395948421210051 Non-Smoking with probability 99.37450289726257 

Tabel 7. Hasil Pengujian dengan Status Salah Prediksi dengan Algoritma *EfficientNetV2M*

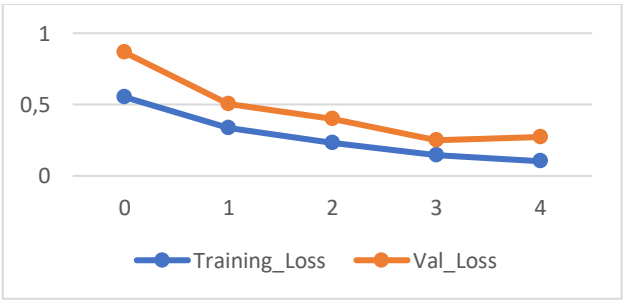
No	Pengujian	Gambar
1	*Actual: smoking* *Predicted: non-smoking*	
2	*Actual: non-smoking* *Predicted: smoking*	
3	*Actual: non-smoking* *Predicted: smoking*	
4	*Actual: non-smoking* *Predicted: smoking*	

No	Pengujian	Gambar
5	*Actual: smoking* *Predicted: non-smoking*	
6	*Actual: non-smoking* *Predicted: smoking*	
7	*Actual: non-smoking* *Predicted: smoking*	

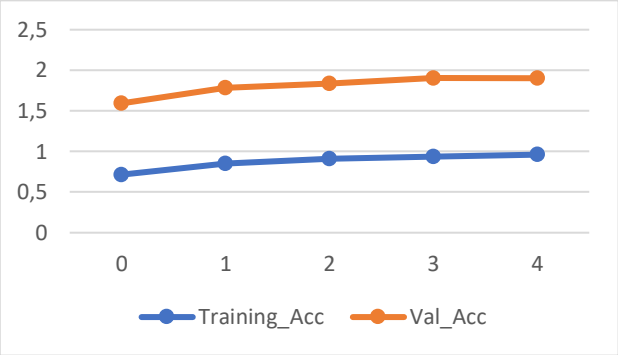
Pada model *Hybrid ViT XGBoost* penulis menggunakan model *custom ViT (Vision Transformer)* untuk tahap ekstraksi fitur, kemudian menerapkan *PCA* untuk masalah dimensi, dan terakhir menggunakan model *XGBoost* untuk tahapan klasifikasi.

Tabel 8. *Pretrained ViT Model Feature Extractor*

Layer (type)	***Output Shape***	***Param #***
input_2 (InputLayer)	*[(None, 224, 224, 3)]*	*0*
vit-b16 (Functional)	*(None, 768)*	*85798656*
flatten (Flatten)	*(None, 768)*	*0*
the_feature_layer (Dense)	*(None, 64)*	*49216*
dense (Dense)	*(None, 16)*	*1040*
dense_1 (Dense)	*(None, 2)*	*34*



Gambar 17. Grafik *Plot* untuk *Training* dan *Validation Loss* pada Algoritma *Hybrid ViT XGBoost*



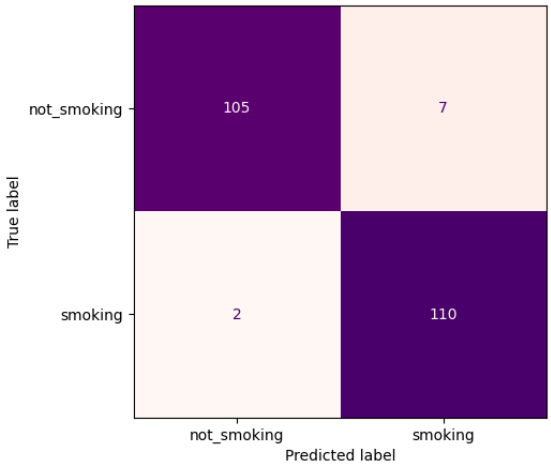
Gambar 18. Grafik *Plot* untuk *Training* dan *Validation Accuracy* pada Algoritma *Hybrid ViT XGBoost*

Tabel 9. *Dimensionality Reduction ViT Model dengan* Menggunakan *PCA 21 components*

	feature_1	feature_2	feature_3	feature_4	feature_5
0	-2.983104	-0.888609	-0.451803	1.302944	1.923158
1	-3.820315	-1.333465	1.904194	-0.447506	1.921710
2	-3.593379	0.111182	-0.307717	0.602026	1.115056
3	6.521454	0.704257	-0.704859	1.100291	-0.785313
4	6.189744	-1.592988	3.201109	-0.735933	-2.751976
	feature_6	**feature_7**	**feature_8**	**feature_9**	**feature_10**
0	-0.316425	-1.092444	-1.523088	2.484431	0.038066
1	0.525290	0.185757	1.856056	-1.192826	1.761079
2	0.449868	0.572691	0.340225	-1.266885	0.958813
3	2.408743	-0.928984	0.236214	-0.103791	1.143586
4	0.411577	1.225524	2.056109	-1.332366	0.293548
	…	…	…	…	…
	feature_41	**feature_42**	**feature_43**	**feature_44**	**feature_45**
0	0.034545	-0.777752	-0.207807	-0.172206	-0.236020
1	0.308991	0.166270	-0.021222	-0.408598	0.119494
2	0.105139	-0.490532	-0.029649	-0.105723	0.314011
3	-0.412999	0.046208	-0.393806	-0.575268	-0.215666
4	-0.847566	0.085418	-0.350967	0.195639	-0.110429
	feature_46	**feature_47**	**feature_48**	**feature_49**	**feature_50**
0	0.456884	-0.294025	-0.008715	0.214049	0.184988
1	-0.533594	-0.224619	-0.635104	0.168260	-0.246595
2	-0.572049	-0.342375	-0.256743	0.602142	1.174022
3	0.854613	0.343095	-0.087747	0.128650	-0.182113
4	0.038047	0.530518	-0.477010	-0.107689	0.094952

Tabel 10. Laporan Klasifikasi dengan Algoritma *Hybrid ViT XGBoost*

Classification Report	***Precision***	***Recall***	***F1-Score***	***Support***
Not smoking	*0,98*	*0,94*	*0,96*	*112*
Smoking	*0,94*	*0,98*	*0,96*	*112*
Accuracy			*0,96*	*224*
Macro avg	*0,96*	*0,96*	*0,96*	*224*
Weighted avg	*0,96*	*0,96*	*0,96*	*224*



Gambar 19. *Confusion Matrix* dengan Algoritma *Vision Transformer*

Tabel 11. Hasil Pengujian dengan dengan Algoritma *Vision Transformer*

No	Pengujian	Gambar
1	*Actual: smoking* *Predicted: smoking*	
2	*Actual: not_smoking* *Predicted: not_smoking*	
3	*Actual: not_smoking* *Predicted: not_smoking*	
4	*Actual: not_smoking* *Predicted: not_smoking*	

No	Pengujian	Gambar
5	*Actual: smoking* *Predicted: smoking*	
6	*Actual: not_smoking* *Predicted: not_smoking*	
7	*Actual: smoking* *Predicted: not_smoking*	
8	*Actual: not_smoking* *Predicted: not_smoking*	
9	*Actual: smoking* *Predicted: smoking*	
10	*Actual: not_smoking* *Predicted: not_smoking*	

4. KESIMPULAN

Penelitian yang disajikan menunjukkan bahwa metode Vision Transformer menunjukkan peningkatan kinerja dalam hal akurasi dan waktu eksekusi yang terbaik dibandingkan dengan metode lainnya, dengan hasil bahwa metode EfficientNetV2M dalam bidang deteksi merokok dengan waktu eksekusi 12 menit dan tingkat akurasi mencapai 97%, pada model MobileNetV3L penulis mendapatkan waktu eksekusi yang lebih cepat yaitu 4 menit dengan akan tetapi dengan tingkat akurasi lebih rendah yaitu 87%, sedangkan pada model Vision Transformer penulis mendapatkan hasil yang sangat memuaskan yaitu waktu eksekusi hanya 4 menit dengan tingkat akurasi 96%.

Saran untuk penelitian selanjutnya adalah disisi teoretis memiliki arah penting untuk penelitian masa depan yaitu usulan konvolusional blok memiliki properti unik yang memungkinkan untuk memisahkan ekspresivitas jaringan (dikodekan oleh lapisan ekspansi) dari kapasitasnya (dikodekan oleh *bottleneck input*).

5. DAFTAR PUSTAKA

[1] A. Khan, S. Khan, B. Hassan, and Z. Zheng, "CNN-Based Smoker Classification and Detection in Smart City Application," *Sensors*, vol. 22, no. 3, Feb. 2022, doi: 10.3390/s22030892.

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018.

[3] E. Eka Citra, D. Hatta Fudholi, and C. Kusuma Dewa, "JURNAL MEDIA INFORMATIKA BUDIDARMA Implementasi Arsitektur EfficientNetV2 Untuk Klasifikasi Gambar Makanan Tradisional Indonesia," 2023, doi: 10.30865/mib.v7i2.5881.

[4] G. Marques, D. Agarwal, and I. de la Torre Díez, "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network," *Applied Soft Computing Journal*, vol. 96, Nov. 2020, doi: 10.1016/j.asoc.2020.106691.

[5] A. F. Anavyanto, M. Maimunah, M. R. A. Yudianto, and P. Sukmasetya, "EfficientNetV2M for Image Classification of Tomato Leaf Deseases," *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, vol. 11, no. 1, pp. 55–76, Mar. 2023, doi: 10.33558/piksel.v11i1.5925.

[6] S.-J. Lee *et al.*, "Early detection of tongue cancer using a convolutional neural network and evaluation of the effectiveness of EcientNet," 2022, doi: 10.21203/rs.3.rs-1628071/v1.

[7] Z. Wang, L. Lei, and P. Shi, "Smoking behavior detection algorithm based on YOLOv8-MNC," *Front Comput Neurosci*, vol. 17, 2023, doi: 10.3389/fncom.2023.1243779.

[8] B. Wang, "Automatic Mushroom Species Classification Model for Foodborne Disease Prevention Based on Vision Transformer," *J Food Qual*, vol. 2022, 2022, doi: 10.1155/2022/1173102.

[9] L. Papa, P. Russo, I. Amerini, and L. Zhou, "A Survey on Efficient Vision Transformers: Algorithms, Techniques, and Performance Benchmarking," *IEEE Trans Pattern Anal Mach Intell*, 2024, doi: 10.1109/TPAMI.2024.3392941.

[10] M. Raichura, N. Chothani, and D. Patel, "Efficient CNN-XGBoost technique for classification of power transformer internal faults against various abnormal conditions," *IET Generation, Transmission and Distribution*, vol. 15, no. 5, pp. 972–985, Mar. 2021, doi: 10.1049/gtd2.12073.

[11] N. Lin, J. Fu, R. Jiang, G. Li, and Q. Yang, "Lithological Classification by Hyperspectral Images Based on a Two-Layer XGBoost Model, Combined with a Greedy Algorithm," *Remote Sens (Basel)*, vol. 15, no. 15, Aug. 2023, doi: 10.3390/rs15153764.

[12] R. Hussein, S. Lee, and R. Ward, "Multi-Channel Vision Transformer for Epileptic Seizure Prediction," *Biomedicines*, vol. 10, no. 7, Jul. 2022, doi: 10.3390/biomedicines10071551.

[13] Y. Habchi *et al.*, "Machine Learning and Vision Transformers for Thyroid Carcinoma Diagnosis: A review," Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.13843

[14] I. A. Esha, "Multiclass Emotion Classification by using Spectrogram Image Analysis: A CNN-XGBoost Fusion Approach," 2023.

[15] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," Apr. 2021, [Online]. Available: http://arxiv.org/abs/2104.00298

[16] H. Zhang *et al.*, "ResNeSt: Split-Attention Networks," 2020.

[17] A. Howard *et al.*, "Searching for MobileNetV3," 2019.

[18] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Tricks for Image Classification with Convolutional Neural Networks," Dec. 2018, [Online]. Available: http://arxiv.org/abs/1812.01187

[19] B. Graham *et al.*, "LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference," Apr. 2021, [Online]. Available: http://arxiv.org/abs/2104.01136

[20] M. Ding, B. Xiao, N. Codella, P. Luo, J. Wang, and L. Yuan, "DaViT: Dual Attention Vision Transformers," 2022. [Online]. Available: https://github.com/microsoft/DaViT.

[21] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-Scale Conv-Attentional Image Transformers," 2021. [Online]. Available: https://github.com/mlpc-ucsd/CoaT.

[22] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.11929

[23] A. Khan, S. Khan, B. Hassan, R. Khan, and Z. Zheng, "SmokerViT: A Transformer-Based Method for Smoker Recognition," *Computers, Materials and Continua*, vol. 77, no. 1, pp. 403–424, 2023, doi: 10.32604/cmc.2023.040251.

[24] I. Mudzakir and T. Arifin, "Klasifikasi Penggunaan Masker dengan Convolutional Neural Network Menggunakan Arsitektur MobileNetv2," *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, vol. 12, no. 1, p. 76, Jun. 2022, doi: 10.36448/expert.v12i1.2466.

[25] M. Ichwan and S. Hadi, "MIND (Multimedia Artificial Intelligent Networking Database Kinerja Model EfficientNetV2M dalam Klasifikasi Citra Tutupan dan Penggunaan Lahan," *Journal MIND Journal | ISSN*, vol. 8, no. 2, pp. 203–216, 2023, doi: 10.26760/mindjournal.v8i2.203-216.