

Kinerja Model EfficientNetV2M dalam Klasifikasi Citra Tutupan dan Penggunaan Lahan

MUHAMMAD ICHWAN, IHSAN SAIFUL HADI

Program Studi Informatika, Institut Teknologi Nasional Bandung
Email: ichwan@itenas.ac.id

Received 14 Juli 2023 | *Revised* 29 Agustus 2023 | *Accepted* 29 September 2023

ABSTRAK

Tutupan lahan merujuk pada unsur-unsur fisik yang terlihat di permukaan bumi, termasuk berbagai material seperti rumput, aspal, pepohonan, tanah terbuka, air, dan material fisik lainnya. Penggunaan lahan mengacu pada bagaimana manusia memanfaatkan lahan, baik di wilayah perkotaan maupun pedesaan. Klasifikasi menggunakan deep learning merupakan metode yang akurat dan bertujuan untuk membantu mendapatkan informasi mengenai tutupan lahan di suatu wilayah. Pada penelitian ini digunakan metode EfficientNetV2M yang merupakan versi terbaru dari pendahulunya yaitu EfficientNet, yang mana metode EfficientNetV2M diharapkan mampu melakukan tugasnya untuk klasifikasi citra satelit tutupan lahan. Beberapa hyperparameter yang digunakan untuk membantu EfficientNetV2M dalam mencapai kinerja yang optimal ini adalah activation function ReLU dan GELU. Hasil klasifikasi tertinggi yang dapat dicapai oleh EfficientNetV2M ini adalah 97% dengan menggunakan ReLU Activation Function dan Optimizer Adamax.

Kata kunci: *Deep Learning, EfficientNetV2M, tutupan lahan, hyperparameter, activation function.*

ABSTRACT

Land cover refers to the physical elements visible on the Earth's surface, including materials such as grass, asphalt, trees, bare soil, water, and other physical materials. Land use refers to how humans utilize land, both in urban and rural areas. Classification using deep learning is an accurate method and aims to help obtain information about land cover in an area. In this study, the EfficientNetV2M method was used which is the latest version of its predecessor, namely EfficientNet, which method is expected to be able to carry out operations for land cover satellite image classification. Some of the hyperparameters used to help EfficientNetV2M achieve optimal performance are the ReLU and GELU activation functions. The highest classification result that can be achieved by EfficientNetV2M is 97% by using ReLU Activation Function and Optimizer Adamax.

Keywords: *Deep Learning, EfficientNetV2M, land cover, hyperparameter, activation function.*

1. PENDAHULUAN

Pada penelitian sebelumnya berhasil melakukan klasifikasi citra tutupan lahan menggunakan *dataset EuroSat* dan menggunakan metode *Xception* yang menghasilkan nilai akurasi yaitu 96,6% dengan *activation function SELU (Scaled Exponential Linear Unit)*, nilai akurasi 96% dengan menggunakan *activation function ReLU* dan nilai akurasi 95,9% dengan menggunakan *activation function Swish (Utama, 2021)*. Penelitian tersebut merupakan wujud perhatian terhadap perkembangan pada suatu wilayah pada permukaan bumi.

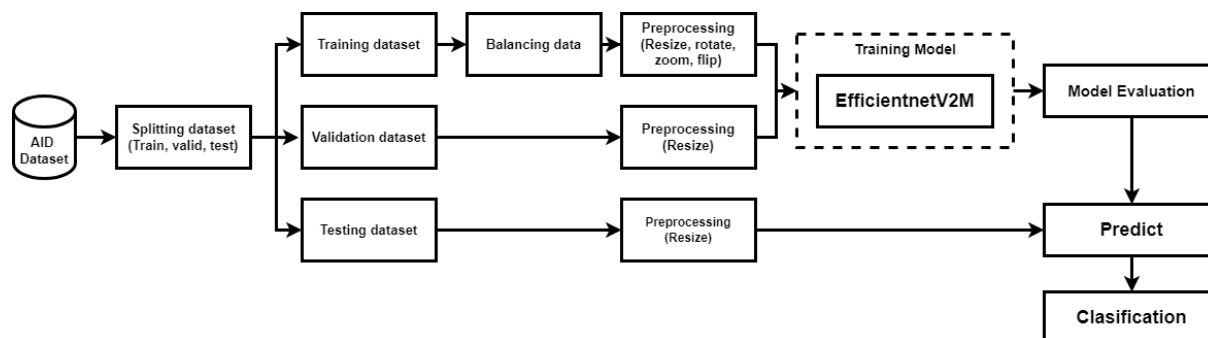
Perkembangan pada suatu wilayah akan menghasilkan perubahan yang dapat diamati secara visual. Perubahan fisik tersebut terjadi karena adanya perkembangan lahan yang dipengaruhi oleh faktor alam dan manusia seperti perkebunan, pertanian, reklamasi lahan dan pembangunan infrastruktur yang mana dapat membantu suatu penilaian dari perkembangan suatu wilayah, upaya konservasi lahan dan perencanaan pembangunan pada wilayah tersebut **(Utama, 2021)**. Untuk mengarahkan peruntukan lahan secara tepat di wilayah yang mengalami perkembangan, diperlukan sebuah perencanaan yang baik. Dalam hal ini, diperlukan suatu metode yang akurat dan efektif untuk mendapatkan informasi mengenai tutupan lahan di wilayah tersebut. Salah satu teknologi yang efektif dalam pemetaan tutupan lahan adalah teknologi penginderaan jauh **(Kusuma, Purwadi, & Marcos, 2021)**. Dalam konteks ini, tutupan lahan menjadi penanda material fisik permukaan bumi, mencakup berbagai elemen permukaan bumi **(Sampurno & Thoriq, 2016)**.

Tutupan lahan meliputi segala jenis penghijauan seperti hutan, pegunungan, padang rumput dan sebagainya. Tutupan lahan dapat mengalami perubahan akibat aktivitas manusia contohnya perkebunan, pertanian, danau atau kawasan pemukiman seperti perumahan, industri dan bangunan tinggi, yang bisa disebut dengan penggunaan lahan. Salah satu pemanfaatan penginderaan jauh menggunakan *machine learning* adalah untuk memprediksi tanaman pangan menggunakan citra satelit, perencanaan pembangunan infrastruktur dan pembuatan saluran pengairan untuk penduduk setempat **(Triscowati & Wijayanto, 2019)**. Teknologi *Machine Learning* saat ini memberikan manfaat yang signifikan dalam berbagai aspek, termasuk dalam pengklasifikasian jenis-jenis lahan. Keuntungan utamanya adalah penghematan waktu dan energi dibandingkan dengan melakukan pengklasifikasian secara langsung di lapangan **(Sari, Baskara, Pratama, & Faidhorrahman, 2021)**. Salah satu *subset* atau cabang dari *machine learning* adalah *deep learning* yang telah menjadi tren yang tumbuh paling cepat dalam analisis data besar dan telah diterapkan secara luas dan berhasil ke berbagai bidang seperti *natural language processing*, *image classification* dan *speech enhancement*. Dikarenakan kinerjanya yang luar biasa dibandingkan dengan algoritma *traditional learning (Li, Zhang, Xue, Jiang, & Shen, 2018)*.

Pada penelitian ini akan menggunakan sebuah metode *deep learning*, yaitu *EfficientNetV2M* yang merupakan iterasi kedua dari versi terdahulunya yaitu *EfficientNet*. *EfficientNetV2* merupakan keluarga jaringan konvolusional baru yang memiliki kecepatan pelatihan lebih cepat dan efisiensi parameter lebih baik dibandingkan dengan model sebelumnya. Model ini dikembangkan melalui pencarian dalam ruang pencarian yang diperkaya dengan operasi baru seperti *Fused-MBConv (Tan & Le, 2021)*. Pada penelitian ini juga dilakukan perubahan *hyperparameter* yang digunakan daripada penelitian sebelumnya dan akan menggunakan *activation function ReLU* dan *GELU* sebagai pembanding, karena *GELU* dapat digunakan sebagai alternatif *ReLU* yang lebih kompleks. Untuk keperluan *dataset*, akan digunakan *dataset* yang berbeda dari penelitian yang sebelumnya dengan menggunakan AID (*Aerial Image Dataset*). *Dataset* ini juga memiliki kelas yang lebih banyak, yaitu 30 kelas dengan jumlah keseluruhan citra adalah 10.000 citra.

2. METODE PENELITIAN

Pada penelitian ini, proses *training* model EfficientnetV2M menggunakan *hyperparameter* yang telah ditentukan. Dalam pengerjaannya, terdapat tambahan variasi *hyperparameter* yang digunakan. Dikarenakan kebutuhan penelitian lebih mendalam atau sebagai pembandingan. Berikut adalah beberapa proses yang dilakukan pada proses *training* model EfficientnetV2M dan proses pengujianya, divisualisasikan dalam bentuk diagram blok berikut.

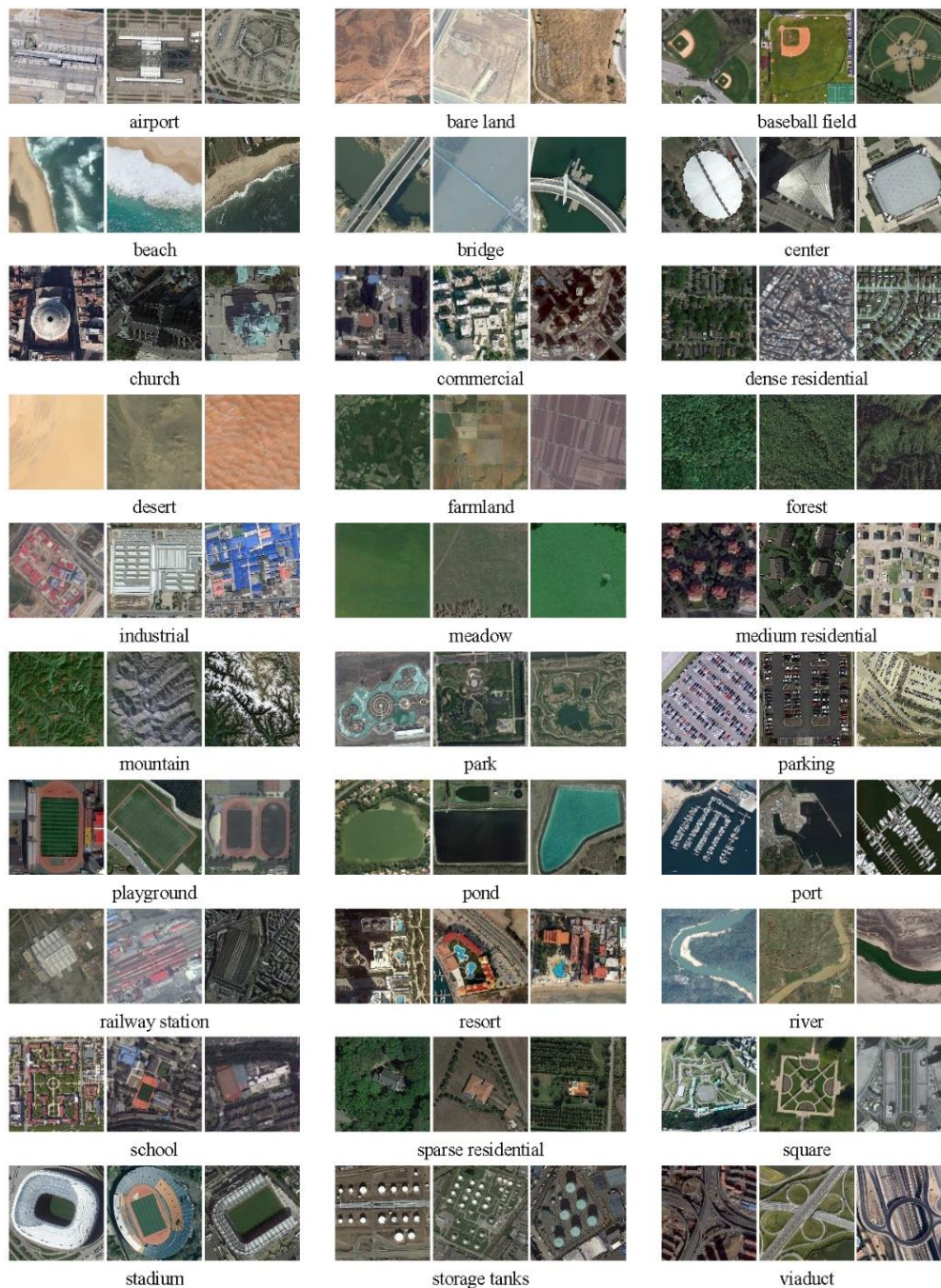


Gambar 1. Block Diagram Penelitian

Pada Gambar 1 di atas, proses sistem diawali dengan *dataset* yang dilakukan proses *splitting dataset*, yang memiliki tujuan untuk membagi dataset menjadi tiga himpunan data yaitu *training*, *testing* dan *validation* data. Data *training* merupakan data citra yang akan dilatih dengan menggunakan metode *EfficientNetV2M*, namun sebelumnya data *training* akan dilakukan *balancing* data yang dimaksudkan untuk menyeimbangkan jumlah data yang terdapat pada masing-masing kelas *dataset*, dan penting untuk efisiensi pembelajaran dan untuk pengambilan keputusan yang tidak *bias* (Susan & Kumar, 2021). Kemudian data *training* dan data *validation* akan dilakukan *preprocessing*, yaitu *resize*, *rotate*, *flip*, dan *zoom* khusus untuk data *training* dan *resize* untuk data *validation*. Lalu data *training* akan dilakukan *training* dengan menggunakan *EfficientNetV2M* yang dikombinasikan dengan *hyperparameter* seperti *optimizer*, *activation function*, *epoch* dan *learning rate* yang sudah ditentukan. Setelah selesai melakukan *training*, *output* dari proses ini akan menghasilkan tingkat akurasi, presisi, *recall*, *F1-Score* dan lama waktu yang dibutuhkan dalam melakukan *training*. Lalu, model yang dihasilkan akan dilakukan pengujian dengan data *testing* untuk klasifikasi sekaligus untuk membuktikan tingkat akurasi yang dihasilkan dengan mengujinya langsung menggunakan data selain data *training* dan data *validation*.

2.1. Dataset

Dataset yang digunakan merupakan kumpulan citra penginderaan jauh yang dikumpulkan oleh Gui-Song Xia dan dapat diakses melalui *Kaggle*. Terdapat sejumlah 10.000 citra yang terbagi kedalam 30 jenis atau kelas.



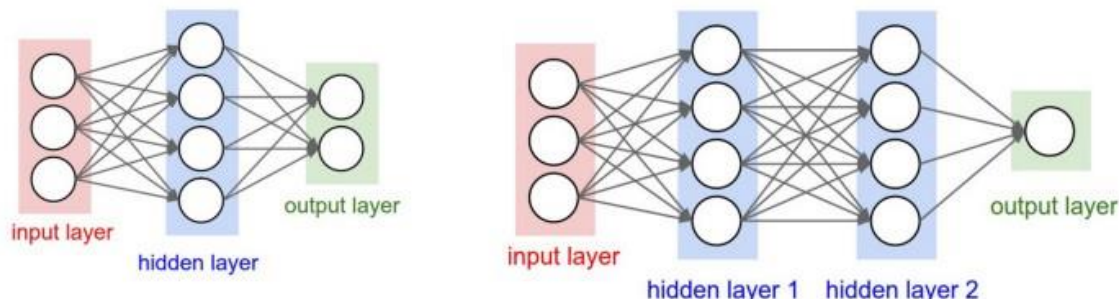
Gambar 2. Sampel Citra *Dataset*

Gambar 2 adalah sampel dari AID (*Aerial Image Dataset*) yang merupakan sebuah *dataset* yang termasuk memiliki berskala besar untuk klasifikasi penginderaan jauh, yaitu 600x600 (Xia, et al., 2017).

2.2. *Deep Learning*

Deep learning adalah teknik *machine learning* yang menggunakan jaringan *neural* yang terdiri dari banyak lapisan untuk mempelajari pola-pola yang terkandung dalam data. Jaringan *neural* ini terdiri dari *node* atau *unit* yang terhubung satu sama lain dan dapat memproses dan menyimpan informasi. Contoh penggunaan *deep learning* adalah untuk menyelesaikan tugas-

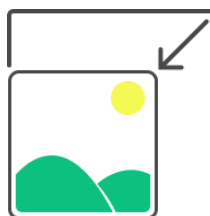
tugas seperti pengenalan wajah, pengenalan suara, dan pemahaman bahasa. Saat ini *deep learning* merupakan pendekatan *machine learning* canggih baru yang sangat diminati baik dalam penelitian akademik maupun aplikasi industri (Nguyen, Kieu, & Cai, 2018). Gambar 3 merupakan contoh visualisasi dari arsitektur jaringan saraf.



Gambar 3. Arsitektur Jaringan Saraf

2.3. Preprocessing dataset

Preprocessing merupakan salah satu tahap dalam pemrosesan data agar data dapat layak digunakan untuk tahap pemrosesan berikutnya (Edha, Sitorus, & Ristian, 2020). Hal ini juga bertujuan untuk menyesuaikan kebutuhan ekstraksi fiturnya dengan citra yang digunakan. *Dataset* yang digunakan akan melalui proses *resize* untuk menurunkan tingkat resolusi citra. Citra yang terdapat didalam *dataset* yang digunakan memiliki resolusi 600x600 akan diturunkan resolusinya menjadi 200x200 agar proses *training* dapat dilakukan lebih singkat saat melakukan proses *training*.



Gambar 4. Ilustrasi Proses *Resize*

Gambar 4 di atas merupakan ilustrasi proses *resize*. Menurunkan skala resolusi sebuah citra dalam melakukan *training dataset* dapat meningkatkan efisiensi waktu saat melakukan pelatihan *dataset* (Saponara & Elhanashi, 2021).

2.4. Splitting Dataset

Dalam *deep learning splitting data* muncul ketika data yang diberikan dibagi menjadi dua atau lebih himpunan bagian, sehingga model dapat dilatih, diuji dan dievaluasi. *Splitting dataset* memiliki tujuan untuk membagi jumlah citra *dataset* menjadi kumpulan citra *training*, *testing* dan *validation*. Pada penelitian ini rasio pembagiannya adalah 90% jumlah citra *training*, 5% jumlah citra *testing* dan 5% jumlah citra *validation*. Tabel 1 di bawah ini merupakan jumlah data pada set data latih, set data validasi dan set data uji.

Tabel 1. Dataset

No	Jenis Lahan	Data Latih	Data Validasi	Data Uji
1	<i>Airport</i> (Bandara)	324	18	18
2	<i>Bare Land</i> (Lahan kosong)	279	16	15
3	<i>Baseball Field</i> (Lap. Baseball)	198	11	11
4	<i>Beach</i> (Pantai)	360	20	20
5	<i>Bridge</i> (Jembatan)	324	18	18
6	<i>Center</i> (Pusat)	234	13	13
7	<i>Church</i> (Gereja)	216	12	12
8	<i>Commercial</i> (Komersial)	315	17	18
9	<i>Dense Residential</i> (Pemukiman padat)	369	20	21
10	<i>Desert</i> (Gurun)	270	15	15
11	<i>Farmland</i> (Pertanian)	333	18	19
12	<i>Forest</i> (Hutan)	225	13	12
13	<i>Industrial</i> (Industrial)	351	20	19
14	<i>Meadow</i> (Padang rumput)	252	14	14
15	<i>Medium Residential</i> (Pemukiman Sedang)	261	14	15
16	<i>Mountain</i> (Gunung)	306	17	17
17	<i>Park</i> (Taman)	315	18	17
18	<i>Parking</i> (Tempat Parkir)	351	20	19
19	<i>Playground</i> (Taman Bermain)	333	19	18
20	<i>Pond</i> (Kolam)	378	21	21
21	<i>Port</i> (Pelabuhan)	342	19	19
22	<i>Railway Station</i> (Stasiun Kereta)	234	13	13
23	<i>Resort</i>	261	15	14
24	<i>River</i> (Sungai)	369	20	21
25	<i>School</i> (Sekolah)	270	15	15
26	<i>Sparse Residential</i> (Pemukiman jarang)	270	15	15
27	<i>Square</i> (Lapangan)	297	16	17
28	<i>Stadium</i> (Stadion)	261	14	15
29	<i>Storage Tanks</i> (Tangki penyimpanan)	324	18	18
30	<i>Viaduct</i>	378	21	21

2.5. Activation Function

Activation function atau fungsi aktivasi pada CNN biasanya digunakan setelah proses konvolusi (*convolution*), di mana hasil dari proses konvolusi tersebut merupakan fitur yang terdeteksi dari sebuah citra. *Activation function* memiliki fungsi untuk menghitung nilai keluaran berdasarkan nilai *input* dan bobot pada *neuron* (Wibawa, 2017).

Pada penelitian ini akan menggunakan *activation function*, yaitu *ReLU* dan *GELU* yang diharapkan dapat menghasilkan hasil akhir model yang optimal.

2.5.1 ReLU Activation Function

Rectified Linear Unit merupakan salah satu *activation function* yang sering kali ditemukan pada model *deep learning*, dan merupakan salah satu yang paling populer, karena diketahui dapat mengurangi waktu pelatihan (Baldassi, Malatesta, & Zecchina, 2019). ReLU memiliki persamaan yang ditunjukkan pada Persamaan (1).

$$ReLU(x) = \max(0, x) \tag{1}$$

Jadi, jika nilai x adalah positif atau nol, maka nilai *output* ReLU akan sama dengan x . Namun, jika nilai x adalah negatif, maka nilai *output* ReLU akan menjadi 0. Dengan kata lain, ReLU memotong nilai-nilai negatif menjadi 0, sementara nilai-nilai positif atau nol tetap tidak berubah.

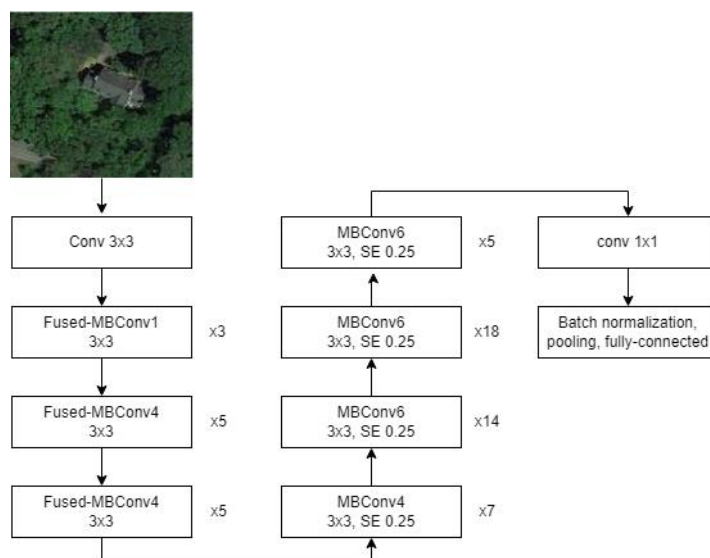
2.5.2 GELU Activation Function

Gaussian Error Linear Units merupakan salah satu alternatif *activation function* selain *ReLU*. Fungsi aktivasi *GELU* hadir sebagai alternatif yang halus dan dapat dihitung turunan untuk menggantikan fungsi aktivasi *ReLU* yang umum digunakan, tanpa mengurangi manfaat yang dimiliki oleh *ReLU*. *GELU* dapat dirumuskan sebagai Persamaan (2) (Lee, 2023).

$$GELU(x) = 0.5x \left(\tanh \left[\sqrt{\frac{2}{\pi}} (x + 0.044715x^3) \right] \right) \tag{2}$$

2.6. EfficientNetV2M

EfficientnetV2 merupakan pengembangan dari versi terdahulunya, yaitu *EfficientNet*. Perbedaan paling terlihat adalah dari struktur arsitekturnya, dengan tambahan sebuah operasi baru, yaitu *Fused-MBConv*. Gambar 5 di bawah merupakan arsitektur *EfficientNetV2M*.



Gambar 5. Arsitektur *EfficientNetV2M*

Metode *EfficientNetV2M* ini dikombinasikan dengan *hyperparameter* sehingga diharapkan dapat menghasilkan nilai akurasi dan kinerja yang baik pada saat proses *training*. Beberapa *optimizer* yang digunakan adalah *Adam*, *Adamax* dan *RMSprop*. Dan menggunakan *Activation function* yang berbeda yaitu, *ReLU* dan *GELU*.

2.7. Confusion matrix

Confusion matrix merupakan sebuah tabel yang digunakan untuk mengevaluasi kinerja dari sebuah algoritma klasifikasi. Ini adalah cara untuk menyimpulkan hasil dari evaluasi klasifikasi, di mana baris mewakili label kelas yang sebenarnya dan kolom mewakili label kelas yang diprediksi. Entri di dalam tabel adalah jumlah dari *instance* yang diprediksi untuk termasuk ke dalam kelas tertentu. Berikut adalah contoh dari Tabel 2 *Confusion Matrix*.

Tabel 2. Confusion Matrix

		<i>Actual Values</i>	
		<i>True</i>	<i>False</i>
<i>Predicted Values</i>	<i>True</i>	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
	<i>False</i>	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Dimana:

True Positives (TP) merupakan jumlah data positif yang diklasifikasikan dengan benar oleh sistem.

True Negatives (TN) merupakan jumlah data negatif yang diklasifikasikan dengan benar oleh sistem.

False Negatives (FN) merupakan jumlah data negatif yang diklasifikasikan salah oleh sistem.

False Positives (FP) merupakan jumlah data positif yang diklasifikasikan salah oleh sistem.

Secara umum, akurasi merupakan perbandingan antara data yang diklasifikasikan dengan benar dengan seluruh data. Nilai akurasi dapat dihitung dengan menggunakan Persamaan (3) **(Pratiwi, Handayani, & Sarjana, 2020)** :

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (3)$$

Kemudian terdapat nilai presisi yang menggambarkan jumlah data dengan kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasikan dengan positif, berikut persamaan untuk memperoleh nilai presisi ditunjukkan pada Persamaan (4) **(Pratiwi, Handayani, & Sarjana, 2020)**:

$$\text{Presisi} = \frac{TP}{TP + FP} \times 100\% \quad (4)$$

Lalu terdapat nilai *recall* yang menunjukkan berapa persentase data dengan kategori positif yang terklasifikasi dengan benar oleh sistem, berikut adalah persamaan untuk mengetahui nilai *recall* ditunjukkan pada Persamaan (5) **(Pratiwi, Handayani, & Sarjana, 2020)**:

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \quad (5)$$

Untuk mengetahui indikasi bahwa model klasifikasi yang dibuat mempunyai presisi dan *recall* yang baik dapat dihasilkan melalui nilai *f1-score*, nilai tersebut memiliki nilai terbaik dengan *index* 1.0 dan nilai terburuk sama dengan 0. Berikut adalah persamaan untuk menghasilkan nilai *f1-score* ditunjukkan pada Persamaan (6):

$$\frac{1}{F1} = \frac{1}{2} \left(\frac{1}{\text{Presisi}} + \frac{1}{\text{recall}} \right) \quad (6)$$

3. HASIL DAN PEMBAHASAN

Pada penelitian ini proses *training* model menggunakan metode *EfficientNetV2M* yang dijalankan melalui aplikasi *website* Google Colab. Untuk mengimplementasikan dua *hyperparameter activation function*, yaitu *ReLU*, *GELU*. Dilakukan beberapa percobaan untuk mencari nilai akurasi, presisi, *recall* dan *f1-score* terbaik dengan mengkombinasikan *activation function* bersama dengan beberapa *optimizer* dan *learning rate*. Pada percobaannya dilakukan dengan menggunakan nilai epoch 40 dan 50. Kemudian dengan menggunakan learning rate 0.1, 0.01 dan 0.001. Juga dengan menggunakan optimizer adam, adamax, RMSprop dan SGD. Dalam percobaan tersebut nilai epoch 40 menjadi lebih efektif, karena dalam proses pelatihannya nilai akurasi terbaik sudah dapat dihasilkan pada epoch 21 hingga 37 dengan menggunakan metode *EarlyStop* sehingga proses pelatihan dapat dihentikan pada epoch tertentu ketika sudah mencapai nilai *loss* paling rendahnya dan pada 4 epoch selanjutnya tidak mengalami penurunan nilai *loss*. Hal ini dilakukan untuk menghindari model yang *overfitting*. Kemudian dalam percobaan tersebut menghasilkan nilai learning rate 0.001 dengan nilai akurasi tertinggi, juga optimizer adam, adamax dan RMSprop yang menghasilkan nilai akurasi yang cukup tinggi. Oleh karena itu dilakukan pengujian menggunakan epoch 40 dengan learning rate 0.001 dan optimizer adam, adamax dan RMSprop untuk mencapai nilai akurasi, presisi, *recall* dan *f1-score* yang tinggi dan model yang optimal. Namun pemilihan epoch, learning rate dan optimizer yang digunakan dapat berbeda, tergantung dengan dataset dan arsitektur yang digunakan.

3.1. Pengujian model

Pada pengujian ini dilakukan dengan *hyperparameter epoch* 40, *Optimizer Adam*, *Adamax* dan *RMSprop* dan dengan nilai *Learning Rate* 0.001. Berikut adalah hasil dari pengujian tersebut ditunjukkan pada Tabel 3 dan Tabel 4:

Tabel 3. Pengujian Activation Function ReLU

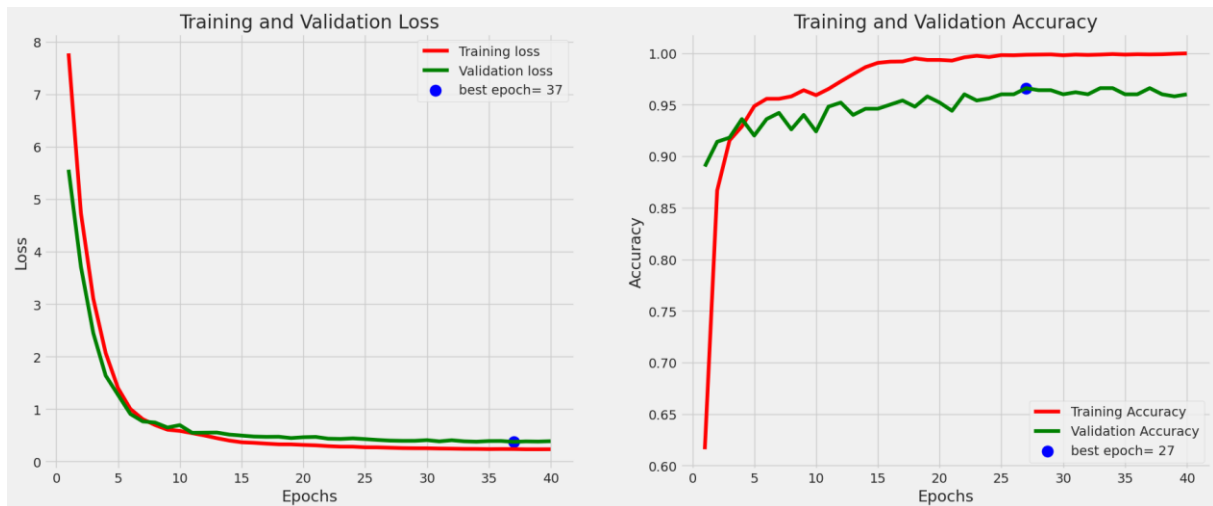
No	Optimizer	Val Acc	Loss	Test Acc	Precision	Recall	F1-Score
1	Adam	0,97	0,36	0,97	0,97	0,96	0,96
2	Adamax	0,96	0,39	0,97	0,97	0,97	0,97
3	RMSprop	0,95	0,28	0,96	0,96	0,95	0,95

Tabel 4. Pengujian Activation Function GELU

No	Optimizer	Val Acc	Loss	Test Acc	Precision	Recall	F1-Score
1	Adam	0,96	0,42	0,95	0,95	0,95	0,94
2	Adamax	0,96	0,43	0,96	0,96	0,96	0,96
3	RMSprop	0,97	0,28	0,96	0,97	0,97	0,97

Hasil dari pengujian dengan menggunakan *activation function ReLU* di atas menghasilkan nilai akurasi paling tinggi yaitu 0,97 dengan menggunakan *optimizer Adamax*. Kemudian hasil dari pengujian dengan menggunakan *activation function GELU* di atas menghasilkan nilai akurasi paling tinggi yaitu 0,97 dengan menggunakan *optimizer RMSprop*. Pada kedua tabel hasil di atas juga dapat dilihat bahwa model dengan *optimizer RMSprop* menghasilkan nilai *loss* paling rendah di antara dua tabel hasil di atas.

Kemudian, berikut adalah grafik akurasi dan *loss* dari salah satu pengujian dengan nilai akurasi tertinggi ditunjukkan pada Gambar 6 dan Gambar 7.



Gambar 6. Grafik Akurasi Dan *Loss* Pengujian Menggunakan *Activation Function* ReLU

Pada grafik yang terdapat pada Gambar 6 diatas, untuk model dengan menggunakan *optimizer* Adamax di atas memperlihatkan bahwa nilai *loss* paling rendah terjadi pada *epoch* ke-37, lalu nilai akurasi terbaik terjadi pada *epoch* ke-27.



Gambar 7. Grafik Akurasi Dan *Loss* Pengujian Menggunakan *Activation Function* GELU

Dari hasil grafik pada Gambar 7, akurasi dan *loss* model dengan menggunakan *optimizer* RMSprop di atas memperlihatkan bahwa nilai *loss* terendah dan nilai akurasi tertinggi terjadi pada *epoch* ke-34.

Pada Gambar 9 di atas merupakan visualisasi hasil kinerja klasifikasi salah satu model *EfficientNetV2M* dengan fungsi aktivasi *GELU* dalam melakukan pengujian prediksi dengan menggunakan data uji. Masih terdapat kesalahan prediksi dari beberapa percobaan contohnya pada prediksi citra yang sebenarnya adalah *School* namun pada sistem memprediksinya sebagai *Baseball Field*.

3.2. Analisis kinerja model

Pada analisis kinerja model ini akan membandingkan hasil kinerja *EfficientNetV2M* untuk kasus klasifikasi citra lahan dari penelitian ini dengan hasil kinerja dari penelitian sebelumnya dengan menggunakan metode *Xception*.

Tabel 5. Perbandingan dengan Hasil Penelitian Sebelumnya

Arsitektur	Dataset	Akurasi
Xception (Swish)	Eurosat	95,9%
Xception (SELU)	Eurosat	96,6%
Xception (ReLU)	Eurosat	96%
EfficientNetV2M (ReLU)	AID	97%
EfficientNetV2M (GELU)	AID	97%

Dengan hasil pada Tabel 5 di atas dapat dilihat bahwa *EfficientNetV2* menghasilkan tingkat akurasi yang lebih tinggi pada *activation function* ReLU. Namun hasil tersebut tidak dapat dibandingkan secara seimbang, dikarenakan penggunaan *dataset* yang berbeda dan *hyperparameter* yang juga berbeda. Oleh karena itu dilakukan kembali percobaan dengan menggunakan metode *xception* dengan parameter yang sama dengan *hyperparameter* yang digunakan pada penelitian ini. Maka diperoleh hasil berikut yang ditunjukkan pada Tabel 6:

Tabel 6. Perbandingan Hasil dengan Activation Function ReLU dan GELU

Activation Function	Optimizer	Val Acc	Loss	Test Acc	Precision	Recall	F1-Score
ReLU	Adam	0,96	0,33	0,96	0,96	0,96	0,96
	Adamax	0,96	0,38	0,96	0,96	0,96	0,96
	RMSprop	0,97	0,24	0,97	0,97	0,97	0,97
GELU	Adam	0,97	0,31	0,96	0,96	0,96	0,96
	Adamax	0,96	0,36	0,97	0,97	0,97	0,97
	RMSprop	0,96	0,25	0,96	0,96	0,96	0,96

4. KESIMPULAN

Pada penelitian ini dihasilkan bahwa *EfficientNetv2M* mampu menghasilkan model klasifikasi dengan menggunakan *activation function* ReLU dengan nilai akurasi 97%, presisi 97%, *recall* 97% dan *f1-score* 97% pada pengujian dengan menggunakan *learning rate* 0.001, *optimizer* Adamax, 40 *epoch* dan pada pengujian model klasifikasi dengan menggunakan *activation function* GELU yang menghasilkan nilai akurasi 96%, presisi 97%, *recall* 97% dan *f1-score* 97% pada pengujian dengan menggunakan *learning rate* 0.001, *optimizer* RMSprop, 40 *epoch*. Penggunaan *activation function* ReLU dan GELU tidak menghasilkan perbedaan tingkat akurasi yang signifikan dan keduanya mampu menghasilkan model yang optimal.

Secara keseluruhan, temuan dari penelitian ini mengindikasikan bahwa model *EfficientNetV2M* menunjukkan kinerja yang lebih unggul dalam melakukan klasifikasi tutupan lahan pada citra satelit, terutama saat menggunakan fungsi aktivasi *ReLU*. Hal ini terbukti dari peningkatan nilai akurasi pada tahap validasi dan pengujian, serta melalui evaluasi matriks yang mencakup

presisi, *recall*, dan nilai *F1-score*. Namun, skenario yang berbeda terjadi ketika fungsi aktivasi *GELU* digunakan; dalam kasus ini, model *Xception* justru menunjukkan hasil yang lebih superior dibandingkan dengan *EfficientNetV2*. Implikasi dari penemuan ini adalah potensi penggunaan model ini untuk memberikan kontribusi signifikan dalam upaya pemetaan dan pemantauan lahan, yang diharapkan dapat dilakukan dengan efisiensi dan akurasi yang lebih tinggi.

DAFTAR RUJUKAN

- Baldassi, C., Malatesta, E. M., & Zecchina, R. (2019). Properties of the Geometry of Solutions and Capacity of Multilayer Neural Network with Rectified Linear Unit Activations. *Physical Review Letters* 123, 170602.
- Edha, H., Sitorus, S. H., & Ristian, U. (2020). Penerapan Metode Transformasi Ruang Warna Hue Saturation Intesity (HSI) Untuk Mendeteksi Kematangan Buah Mangga Harum Manis. *Jurnal Komputer dan Aplikasi*, 1-10.
- Kusuma, B. A., Purwadi, & Marcos, H. (2021). Pelatihan Klasifikasi Tutupan Lahan Sebagai Teknologi Penginderaan Jarak Jauh untuk Pemantauan Lahan Pertanian Di Kabupaten Banyumas. *Community Engagement & Emergence Journal*, 28-35.
- Lee, M. (2023). GELU Activation Function in Deep Learning: A Comprehensive Mathematical Analysis and Performance. *arXiv preprint arXiv:2305.12073*.
- Li, Y., Zhang, H., Xue, X., Jiang, Y., & Shen, Q. (2018). Deep learning for remote sensing image classification: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(6), e1264.
- Nguyen, H., Kieu, L.-M., & Cai, C. (2018). Deep learning methods in transportation domain: a review. *IET Intelligent Transport Systems*.
- Pratiwi, B. P., Handayani, A. S., & Sarjana. (2020). Pengukuran Kinerja Sistem Kualitas Udara Dengan Teknologi WSN Menggunakan Confusion Matrix. *Jurnal Informatika Upgris*.
- Sampurno, R. M., & Thoriq, A. (2016). Klasifikasi Tutupan Lahan Menggunakan Citra Landsat 8 Operational Land Imager (OLI) Di Kabupaten Sumedang. *Jurnal Teknotan Vol. 10 No. 2, November 2016*, 62-71.
- Saponara, S., & Elhanashi, A. (2021). Impact of Image Resizing on Deep Learning Detectors . *International Conference on Applications in Electronics Pervading Industry, Environment and Society*, 10-17.
- Sari, Y., Baskara, A. R., Pratama, F., & Faidhorrahman, M. (2021). Penerapan Arsitektur VGG Untuk Klasifikasi Hutan. *Jurnal Teknologi Informasi Universitas Lambung Mangkurat (JTIULM)*, 6(2), 85-92.

- Susan, S., & Kumar, A. (2021). The balancing trick: Optimized sampling of imbalanced datasets—A brief survey of the recent State of the Art. *Engineering Reports*, 3(4), e12298.
- Tan, M., & Le, Q. (2021). EfficientNetV2: Smaller Models and Faster Training. *International conference on machine learning*, 10096-10106.
- Triscowati, D. W., & Wijayanto, A. W. (2019). Peluang Dan Tantangan Dalam Pemanfaatan Teknologi Penginderaan Jauh Dan Machine Learning untuk Prediksi Data Tanaman Pangan Yang Lebih Akurat. *Seminar Nasional Official Statistics*, Vol. 2019, No. 1, pp. 177-187.
- Utama, A. D. (2021). Deteksi Jenis Tutupan Dan Penggunaan Lahan Menggunakan Convolutional Neural Network Arsitektur Xception. *Itenas Library*, 95.
- Wibawa, M. S. (2017). Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah Epoch Terhadap Performa Jaringan Saraf Tiruan. *Jurnal Sistem dan Informatika (JSI)*, 167-174.
- Xia, G.-S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., & Zhang, L. (2017). AID: A Benchmark Dataset for Performance Evaluation of Aerial Scene Classification. *IEEE Trans. on Geoscience and Remote Sensing*.