

IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORK USING MOBILENETV2 TO DISTINGUISH HUMAN AND ARTIFICIAL INTELLIGENCE PAINTING

Dwi Bagia Santosa^{*1}, Agung Wahana², Wisnu Uriawan³

^{1,2,3}Faculty of Science and Technology, Informatics Engineering, UIN Sunan Gunung Djati Bandung, Indonesia
Email: ¹1207050030@uinsgd.ac.id, ²wahana.agung@uinsgd.ac.id, ³wisnu_u@uinsgd.ac.id

(Article received: October 1, 2024; Revision: October 23, 2025; published: February 24, 2025)

Abstract

The advancement of artificial intelligence technology has had a significant impact on various fields, including painting. Artificial intelligence is now able to create works of art that resemble paintings produced by humans with a high level of detail and complexity. However, this progress has also created new problems in the world of painting, namely the difficulty in distinguishing between works produced by humans and those created by artificial intelligence. This problem has an impact on the originality of the artwork and has implications for aspects of ethics and creativity. This study aims to develop a deep learning model that can classify human and artificial intelligence paintings, and overcome the challenges in distinguishing between the two. The methodology used is the Cross Industry Standard Process for Data Mining (CRISP-DM), with a dataset consisting of 1,000 painting images. The architecture used is MobileNetV2, implemented using TensorFlow to build a Convolutional Neural Network (CNN). Techniques such as data preparation, data labeling, data splitting, resizing, and data augmentation are applied to improve model performance. Six test scenarios were carried out with variations in the learning rate, number of epochs, and freeze or unfreeze configurations on the base model. The results showed that the best model with a learning rate of 0.0001, base model unfreeze, and 5 epochs managed to achieve an accuracy of 97%, without any indication of overfitting or underfitting. This model was then implemented on an Android application in TFLite format, which can predict image classes with a confidence level of 89.98%.

Keywords: Art Painting, Convolutional Neural Network, CRISP-DM, MobileNetV2, TensorFlow.

IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK MENGGUNAKAN MOBILENETV2 UNTUK MEMBEDAKAN HASIL KARYA SENI LUKIS MANUSIA DAN KECERDASAN BUATAN

Abstrak

Kemajuan teknologi kecerdasan buatan (*Artificial Intelligence*) telah membawa dampak signifikan dalam berbagai bidang, termasuk seni lukis. *Artificial intelligence* kini mampu menciptakan karya seni yang menyerupai lukisan yang dihasilkan oleh manusia dengan tingkat detail dan kompleksitas yang tinggi. Namun, kemajuan ini juga menimbulkan masalah baru dalam dunia seni lukis yaitu kesulitan dalam membedakan antara karya yang dihasilkan oleh manusia dan yang dibuat oleh kecerdasan buatan. Permasalahan ini berdampak pada orisinalitas karya seni dan memiliki implikasi pada aspek etika dan kreativitas. Penelitian ini bertujuan untuk mengembangkan model *deep learning* yang dapat mengklasifikasikan lukisan manusia dan kecerdasan buatan, serta mengatasi tantangan dalam membedakan keduanya. Metodologi yang digunakan adalah *Cross Industry Standard Process for Data Mining* (CRISP-DM), dengan dataset yang terdiri dari 1.000 gambar lukisan. Arsitektur yang digunakan adalah MobileNetV2, diimplementasikan menggunakan TensorFlow untuk membangun model *Convolutional Neural Network* (CNN). Teknik seperti *data preparation*, *data labeling*, *data splitting*, *resizing*, dan *data augmentation* diterapkan untuk meningkatkan performa model. Enam skenario pengujian dilakukan dengan variasi pada *learning rate*, jumlah *epoch*, dan konfigurasi *freeze* atau *unfreeze* pada *base model*. Hasil penelitian menunjukkan bahwa model terbaik dengan *learning rate* 0.0001, *base model unfreeze*, dan 5 *epoch* berhasil mencapai akurasi sebesar 97%, tanpa indikasi *overfitting* atau *underfitting*. Model ini kemudian diimplementasikan pada aplikasi Android dalam format TFLite, yang dapat memprediksi kelas gambar dengan tingkat *confidence* sebesar 89,98%.

Kata kunci: Convolutional Neural Network, CRISP-DM, MobileNetV2, Seni Lukis, TensorFlow

1. PENDAHULUAN

Seni lukis adalah salah satu cabang seni rupa yang diwujudkan melalui karya dua dimensi yang diciptakan melalui sapuan warna atau cat yang secara tradisional menggunakan kuas. Seni lukis telah mengalami perkembangan teknis yang pesat, dengan alat dan media yang beragam, termasuk seni lukis digital yang tidak lagi menggunakan pewarnaan pigmen [1]. Kemajuan teknologi, terutama dalam bidang kecerdasan buatan (*Artificial Intelligence*), telah membawa tantangan baru bagi dunia seni lukis [2]. Salah satu perkembangan menarik adalah kemampuan kecerdasan buatan dalam menciptakan lukisan yang sangat mirip dengan karya manusia. Penelitian oleh Alice Xue dalam implementasi *Generative Adversarial Networks* (GAN) menunjukkan bahwa 70% dari 242 partisipan *visual test* salah mengira bahwa lukisan yang mereka lihat adalah hasil karya manusia, melainkan merupakan hasil kecerdasan buatan [3].

Dunia seni lukis yang akhir-akhir ini semakin banyak dibanjiri oleh karya yang dihasilkan oleh kecerdasan buatan menimbulkan masalah baru yang mendasar, yaitu kesulitan manusia untuk membedakan antara lukisan yang dihasilkan oleh kecerdasan buatan dan lukisan yang dihasilkan oleh manusia itu sendiri [4]. Masalah ini tidak hanya mencakup kesulitan manusia dalam membedakan keduanya, tetapi juga berpotensi melibatkan isu-isu baru seperti pembajakan kreativitas dan pelanggaran hak cipta terhadap karya seni seseorang. Perbedaan antara kreativitas manusia dan kemampuan kecerdasan buatan untuk meniru seni manusia menciptakan kebutuhan akan pendekatan yang canggih dalam mengidentifikasi dan membedakan antara karya seni yang berasal dari kecerdasan buatan dan hasil ciptaan manusia. Dalam konteks ini, penting untuk diketahui bahwa kemampuan kecerdasan buatan dalam menghasilkan karya seni lukis yang meniru gaya manusia tidak hanya menghadirkan tantangan teknis, tetapi juga menimbulkan pertanyaan filosofis dan etis. Keaslian dan orisinalitas karya seni lukis menjadi semakin sulit untuk dijaga ketika kecerdasan buatan dapat menghasilkan karya yang tidak dapat dibedakan dari karya manusia. Selain itu, implikasi terhadap pasar seni juga signifikan, dimana kolektor dan penikmat seni mungkin kesulitan menilai nilai intrinsik dari sebuah karya jika tidak dapat dipastikan apakah karya tersebut diciptakan oleh manusia atau kecerdasan buatan.

Computer vision merupakan bidang dalam ilmu komputer yang berkaitan dengan pemrosesan dan analisis gambar atau video. *Computer vision* memiliki algoritma yang dapat menjadi solusi potensial untuk mengidentifikasi atau mengenali perbedaan antara hasil karya seni lukis manusia dan kecerdasan buatan. Algoritma tersebut adalah algoritma *Convolutional Neural Network* (CNN) yang mana memiliki performa yang luar biasa dalam

melakukan identifikasi atau klasifikasi sebuah gambar [5]. Algoritma ini memanfaatkan operasi konvolusi pada lapisan-lapisan dalam arsitekturnya yang dapat memungkinkan untuk mengidentifikasi pola-pola kompleks pada data gambar. Inilah yang menjadikan *Convolutional Neural Network* (CNN) sebagai salah satu faktor kunci dalam kemampuan pengenalan pola pada gambar. *Convolutional Neural Network* (CNN) ini telah banyak digunakan untuk melakukan klasifikasi gambar, diantaranya adalah untuk klasifikasi motif batik solo [6], klasifikasi penyakit daun jagung [7], deteksi sampah metal [8], klasifikasi kanker kulit [9] dan klasifikasi penyakit mata [10].

Adapun beberapa penelitian yang telah dilakukan sebelumnya untuk melakukan klasifikasi lukisan, seperti yang dilakukan oleh Yohannes et al. [11] dalam implementasi CNN dengan membandingkan arsitektur VGG-19 dan ResNet-50 untuk klasifikasi lukisan karya Van Gogh dengan akurasi yang didapat sebesar 93,28% untuk arsitektur VGG-19 dan 90,28% untuk arsitektur ResNet-50. Kemudian penelitian yang dilakukan oleh Richadson et al. [12] dalam membandingkan arsitektur CNN yaitu VGG-16, VGG-19, ResNet, dan Xception untuk klasifikasi pencipta dari lukisan terkenal mendapatkan akurasi sebesar 95% untuk arsitektur Xception, 92% untuk arsitektur ResNet, dan untuk kedua arsitektur VGG hanya mendapatkan akurasi paling besar 60%. Kemudian penelitian yang dilakukan oleh Kaustav dan Anita [13] dalam membedakan lukisan yang berasal dari India dan luar India dengan menggunakan CNN konvensional dan 20 ribu gambar karya seni yang digunakan dalam eksperimennya mendapatkan akurasi sebesar 85,05%.

MobileNetV2 merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang dirancang khusus untuk penggunaan pada perangkat seluler yang memiliki daya komputasi dan memori yang terbatas [14]. Arsitektur ini menggunakan konsep *depthwise separable convolution* yang mana pendekatan ini menggantikan konvolusi standar dengan mengurangi jumlah parameter dan beban komputasi tanpa mengorbankan akurasi secara signifikan [15]. Hal inilah yang membuat model yang dihasilkan oleh arsitektur MobileNetV2 menjadi ringan, sehingga model tersebut dapat beroperasi secara optimal pada perangkat seluler yang memiliki keterbatasan daya komputasi, seperti pada perangkat Android yang digunakan oleh 88% masyarakat di Indonesia [16].

Berdasarkan penjelasan yang telah diberikan di atas, beragam arsitektur *Convolutional Neural Network* telah digunakan untuk melakukan klasifikasi lukisan, namun belum ada penelitian yang menggunakan arsitektur MobileNetV2 untuk klasifikasi lukisan. Penelitian ini berfokus pada implementasi *Convolutional Neural Network* dengan arsitektur MobileNetV2 untuk klasifikasi lukisan

hasil manusia dan hasil kecerdasan buatan, dengan harapan dapat memberikan kontribusi signifikan dalam mengatasi tantangan yang dihadapi dalam membedakan karya seni lukis manusia dan kecerdasan buatan.

2. METODE PENELITIAN

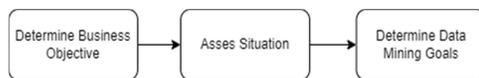
Pada bagian ini dibahas mengenai cara peneliti mengatasi tantangan serta langkah-langkah yang diambil dalam menghadapi permasalahan pada penelitian yang sedang dilakukan. Pada penelitian ini, strategi penyelesaian masalah yang diterapkan adalah melalui pendekatan *Cross Industry Standar Process for Data Mining* (CRISP-DM). CRISP-DM merupakan metodologi yang berorientasi pada proyek *data mining* [17]. Pendekatan ini telah matang dan tetap banyak diterima dalam proyek-proyek *data mining* melalui algoritma *machine learning*. Metodologi ini memberikan pendekatan *lifecycle* dalam proyek-proyek kecerdasan buatan terapan dan dianggap sebagai metodologi yang ideal untuk proses penemuan pengetahuan dalam basis data [18]. CRISP-DM memiliki enam fase proses, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation* dan *deployment* [19]. Fase tersebut dapat dilihat pada Gambar 1.



Gambar 1. Fase CRISP-DM [17]

2.1. Business Understanding

Business understanding merupakan tahap pertama pada metodologi CRISP-DM. Tahap ini berfokus pada beberapa fase seperti pada gambar 2.



Gambar 2. Fase Business understanding

Business understanding merupakan tahap penentuan kriteria *project* dan tujuan bisnis pada penelitian yang dilakukan. Pada tahap ini juga dilakukan eksplorasi terhadap data yang tepat dan andal untuk penelitian yang dilakukan. Kemudian, tahap ini juga menentukan kriteria keberhasilan yang

akan digunakan untuk menilai hasil dari penelitian yang dilakukan.

2.2. Data Understanding

Data understanding merupakan tahap selanjutnya setelah *business understanding* pada metodologi CRISP-DM yang melibatkan eksplorasi awal data untuk memahami karakteristiknya. Fokus pada tahap ini adalah melakukan analisis mendalam terhadap data dan menilai kualitas data.

2.3. Data Preparation

Tahap selanjutnya setelah *data understanding* pada metode CRISP-DM adalah *data preparation*. Tahap ini akan melakukan beberapa langkah *data preprocessing*, yaitu *data labeling*, *data splitting*, *resize*, dan *data augmentation* untuk menyiapkan data sebelum digunakan dalam proses pelatihan model.

2.4. Modeling

Setelah melalui tahapan-tahapan sebelumnya, tahapan selanjutnya adalah *modeling*. Pada tahap ini, mulai memilih model yang sesuai dan merancang eksperimen untuk mengembangkan serta melatih model. Penelitian ini akan menggunakan TensorFlow untuk membangun *Convolutional Neural Network* dengan arsitektur MobileNetV2.

Dalam penelitian ini, pembuatan skenario pengujian dilakukan dengan maksud untuk mencapai hasil optimal atau yang terbaik dari model yang sedang dikembangkan. Terdapat beberapa faktor yang mempengaruhi kinerja sebuah model dalam klasifikasi gambar, salah satunya adalah variasi *learning rate* yang digunakan, melakukan *freeze* atau *unfreeze* pada *base model* dan jumlah *epoch* yang digunakan [20]. Penelitian ini akan melakukan skenario pengujian dengan mempertimbangkan faktor-faktor yang mempengaruhi kinerja model seperti yang sudah dijelaskan sebelumnya. Skenario pengujian dapat dilihat pada tabel 1.

Tabel 1. Skenario Pengujian

Skenario	<i>Base model trainable</i>	<i>Learning rate</i>	<i>Epoch</i>
1	<i>True</i>	0.01	5
2	<i>False</i>	0.01	10
3	<i>True</i>	0.001	5
4	<i>False</i>	0.001	10
5	<i>True</i>	0.0001	5
6	<i>False</i>	0.0001	10

Untuk konfigurasi yang lain seperti *loss*, *metrics*, *batch size* dan *optimizer* model akan menggunakan konfigurasi yang sama. Untuk *loss* menggunakan *categorical_crossentropy*, *metrics* menggunakan *accuracy*, *batch size* berjumlah 32 dan *optimizer* menggunakan *Adam*.

2.5. Evaluation

Pada tahap evaluasi penelitian ini, metode *confusion matrix* digunakan untuk melakukan analisis dan evaluasi terhadap kualitas model yang

telah dikembangkan. *Confusion matrix* menjadi alat visualisasi yang penting dan digunakan untuk menghitung metrik evaluasi, seperti *accuracy*, *precision*, *recall* dan *F1-Score*. *Accuracy* dihitung untuk mengukur sejauh mana model dapat memprediksi dengan benar, lalu *precision* dihitung untuk mengukur sejauh mana prediksi positif model tepat, kemudian *recall* mengukur sejauh mana model dapat mengidentifikasi semua kasus positif yang sebenarnya dan *F1-score* merupakan *harmonic mean* dari *precision* dan *recall*, memberikan gambaran seimbang antara keduanya. Metrik evaluasi tersebut dapat dihitung menggunakan persamaan (1), (2), (3), (4).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F} \quad (1)$$

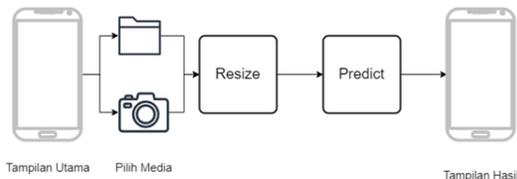
$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$f1 - Score = \frac{2 \times (Precision \times Recall)}{Precision+Re} \quad (4)$$

2.6. Deployment

Tahapan terakhir adalah *deployment*. Pada tahap ini, model terbaik yang sebelumnya telah dipilih akan diintegrasikan ke dalam aplikasi Android menggunakan bahasa pemrograman kotlin. Model yang telah dipilih tersebut akan diubah kedalam format TFLite agar dapat dimplementasikan pada bahasa pemrograman kotlin sehingga dapat digunakan pada aplikasi android. Alur aplikasi pada penelitian ini adalah pertama menampilkan tampilan utama, lalu *user* diberikan pilihan untuk mengambil gambar melalui kamera atau dari galeri, kemudian sistem akan merubah ukuran gambar yang telah *user* pilih sesuai dengan *input size* model yang dibangun, kemudian yang terakhir model akan melakukan prediksi dan sistem menampilkan hasil prediksinya. Gambar 3 merupakan gambaran dari alur aplikasinya.



Gambar 3. Alur aplikasi

3. HASIL DAN PEMBAHASAN

3.1. Business Understanding

Penelitian ini bertujuan untuk mengidentifikasi atau membedakan hasil karya seni lukis manusia dan kecerdasan buatan. Kemajuan teknologi kecerdasan buatan dalam menghasilkan seni lukis membuat manusia sulit membedakan antara lukisan hasil kecerdasan buatan dan manusia itu sendiri. Sulitnya

membedakan atau mengidentifikasi karya lukisan ini dapat memunculkan potensi pembajakan kreativitas dan hak cipta karya seni seseorang.

Untuk mencapai tujuan penelitian ini, diperlukan data yang mewakili karya lukisan hasil manusia dan kecerdasan buatan. Lukisan sendiri terdiri dari beberapa jenis dan gaya, tetapi penelitian ini hanya menggunakan lukisan jenis potret dan *landscape*, jadi data yang digunakan adalah gambar dari masing-masing lukisan jenis potret dan *landscape* hasil ciptaan manusia dan kecerdasan buatan.

Data yang telah dikumpulkan akan digunakan untuk mengembangkan suatu model yang mampu membedakan antara lukisan potret dan *landscape* yang diciptakan oleh manusia dan kecerdasan buatan. Pembangunan model akan menggunakan arsitektur *Convolutional Neural Network (CNN)* dengan *MobileNetV2* dan hasilnya akan diubah ke dalam format TFLite agar dapat diintegrasikan dengan sistem yang akan dikembangkan.

3.2. Data Understanding

Penelitian ini menggunakan data gambar yang terdiri dari karya lukisan jenis potret dan *landscape* hasil kecerdasan buatan dan manusia. Setiap jenis lukisan potret dan *landscape* hasil kecerdasan buatan dan manusia masing-masing memiliki 250 gambar, sehingga total terdapat 1000 gambar. Data gambar yang didapat pada penelitian ini bersumber dari situs leonardo ai, kaggle dan situs lainnya. Data tersebut dikumpulkan dengan maksud untuk mencakup berbagai variasi yang mewakili setiap jenis lukisan secara representatif. Gambar 4 merupakan contoh dari jenis lukisan *landscape* dan potret.

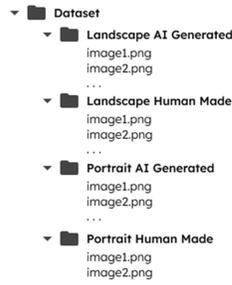


Gambar 4. Contoh jenis lukisan

3.3. Data Preparation

3.3.1. Data Labeling

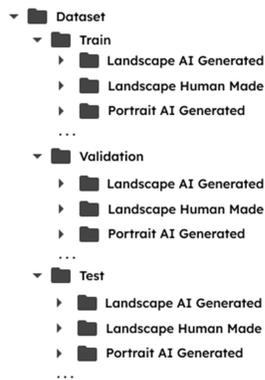
Proses data labeling merupakan tahap pemberian kelas atau kategori untuk setiap data yang terdapat pada dataset. Penelitian ini memiliki 4 kelas yaitu *portrait AI generated*, *portrait human made*, *landscape AI generated* dan *landscape human made*. Pelabelan dilakukan dengan cara membuat sebuah folder dengan nama kelas yang telah dibuat dan mengisi foldernya dengan gambar yang sesuai dengan nama-nama folder tersebut seperti yang dapat dilihat pada gambar 5.



Gambar 5. Direktori data setelah diberi label

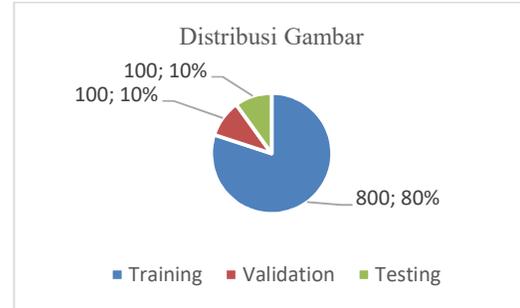
3.3.2. Data Splitting

Data Splitting adalah suatu proses dimana dataset yang telah dikumpulkan dibagi menjadi dua atau lebih subset yang berbeda. Tujuan utama dari *data splitting* adalah untuk memastikan bahwa model yang dikembangkan dapat diuji secara objektif pada data yang belum dilihat sebelumnya. Penelitian ini melakukan *data splitting* dengan 3 subset, yaitu *training*, *validation*, dan *testing*. *Training* data digunakan untuk melatih model itu sendiri. *Validation* data digunakan selama proses pelatihan untuk menyesuaikan parameter model dan mencegah overfitting yang membantu menilai kinerja model pada data yang belum pernah dilihat selama proses latihan. *Test* data Digunakan untuk menguji kinerja akhir model setelah proses pelatihan dan penyesuaian parameter selesai [21]. Gambar 6 merupakan isi dataset setelah dilakukan *data splitting*.



Gambar 6. Direktori dataset setelah data splitting

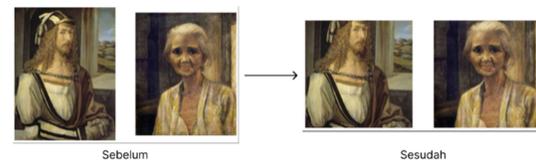
Data training pada penelitian ini berjumlah 200 gambar untuk setiap jenis lukisan potret dan *landscape* hasil kecerdasan buatan dan manusia, sehingga total gambar pada data *training* adalah 800. Lalu *data validation* untuk setiap kelas yang sudah ditentukan berjumlah 25 gambar, sehingga jumlah total *data validation* adalah 100 gambar. Kemudian untuk data *testing* memiliki pembagian yang sama dengan data *validation* yang jumlahnya 100. Distribusi Gambar dapat dilihat pada gambar 7.



Gambar 7. Distribusi dataset

3.3.3. Resize

Penelitian ini melibatkan proses *resizing* untuk mengubah resolusi atau dimensi horizontal dan vertikal pada dataset yang akan digunakan. Proses ini diterapkan agar sesuai dengan kebutuhan sistem, mengingat bahwa arsitektur yang digunakan dalam penelitian ini memerlukan input gambar dengan ukuran 224x224 piksel. Hasil sebelum dan sesudah gambar dilakukan *resizing* dapat dilihat pada gambar 8.

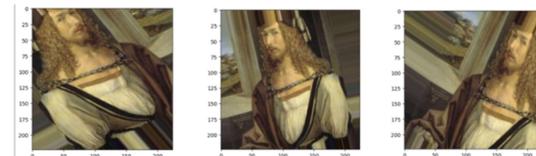


Gambar 8. Hasil *resize*

3.3.4. Data Augmentation

Data augmentation merupakan teknik yang digunakan untuk meningkatkan jumlah dan keragaman data *training* dengan membuat variasi dari data yang sudah ada. Tujuan dari *data augmentation* adalah memberikan keberagaman tambahan pada dataset *training*, sehingga model dapat belajar dari berbagai kondisi dan memperbaiki kinerjanya pada data yang belum pernah dilihat sebelumnya [22].

Penelitian ini melakukan *data augmentation* menggunakan ImageDataGenerator, sebuah class yang tersedia pada library keras untuk melakukan *data augmentation*. ImageDataGenerator pada keras memiliki banyak operasi, namun penelitian ini hanya akan menggunakan beberapa operasi saja yaitu *pixel normalization*, *shearing* dan *zoom in/out*. Contoh hasil data augmentation dapat dilihat pada gambar 9.

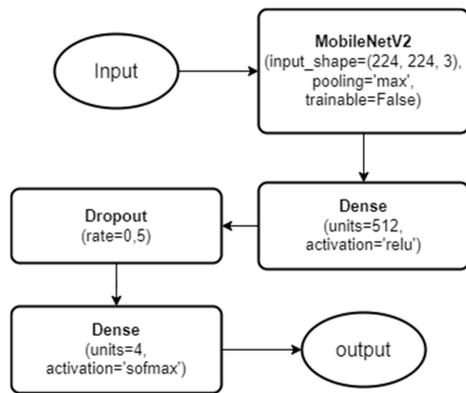


Gambar 9. Hasil *data augmentation*

3.4. Modeling

Pada pembangunan model ini, MobileNetV2 digunakan sebagai *base model*. MobileNetV2 akan berperan sebagai dasar utama dalam *feature*

extractor dari data gambar yang akan diinputkan. Selanjutnya, akan ditambahkan *dense layer* dengan fungsi aktivasi ReLU, yang bertujuan untuk mendeteksi pola dan relasi kompleks dalam data gambar. Kemudian, akan disertakan juga *dropout layer* untuk mengurangi risiko terjadinya *overfitting* pada model yang sedang dikembangkan. Setelah penambahan layer-layer tersebut, model akan dikompilasi dengan menggunakan *optimizer* Adam yang dilengkapi dengan *learning rate* yang bervariasi. Gambar 10 merupakan gambaran dari model yang akan dibangun.



Gambar 10. Arsitektur model yang akan dibangun

3.5. Evaluation

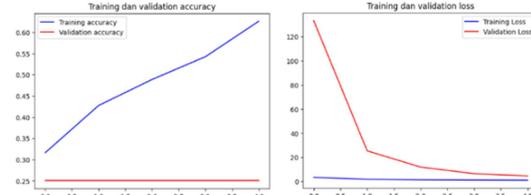
Tahap *evaluation* pada penelitian ini dilakukan guna memastikan model yang dibangun sesuai dengan apa yang telah didefinisikan dalam *business understanding* dan untuk mencari model terbaik yang telah dibangun. Dalam mencari model yang terbaik, penelitian ini melakukan 6 skenario pengujian dengan konfigurasi tersendiri pada setiap skenarionya.

3.5.1. Skenario 1

Pada skenario pertama, model dibangun dengan konfigurasi *unfreeze* pada *base model* serta *learning rate* sebesar 0.01 dan nilai *epoch* 5. Data yang digunakan sebanyak 200 gambar untuk *training* dan 25 gambar untuk *validation*. Pengecekan terhadap *accuracy training* dan *accuracy validation*, serta *loss training* dan *loss validation* dilakukan untuk memonitor apakah model yang dibangun mengalami *underfitting* atau *overfitting*. Hal tersebut sangat berguna dan penting untuk mengetahui bagaimana performa model yang dihasilkan.

Tabel 2. Hasil Skenario 1

Skenario 1	
Accuracy training	62,63 %
Accuracy validation	25 %



Gambar 11. Visualisasi akurasi dan loss skenario 1

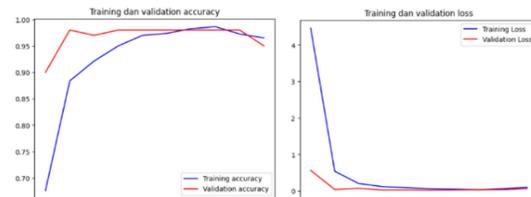
Dari Gambar 11 dapat diketahui bahwa hasil *training* model yang dibangun memiliki performa yang tidak baik dikarenakan *accuracy training* dan *accuracy validation* mendapatkan nilai kurang dari 70%, kemudian perbedaan dari kedua hasil *accuracy* yang cukup jauh. Hal ini menunjukkan bahwa model tersebut tidak baik jika akan digunakan untuk melakukan klasifikasi gambar.

3.5.2. Skenario 2

Pada skenario kedua, model dibangun dengan konfigurasi *freeze* pada *base model* serta *learning rate* sebesar 0.01 dan nilai *epoch* 10. Data yang digunakan sebanyak 200 gambar untuk *training* dan 25 gambar untuk *validation*. Pengecekan terhadap *accuracy training* dan *accuracy validation*, serta *loss training* dan *loss validation* dilakukan untuk memonitor apakah model yang dibangun mengalami *underfitting* atau *overfitting*. Hal tersebut sangat berguna dan penting untuk mengetahui bagaimana performa model yang dihasilkan.

Tabel 3. Hasil Skenario 2

Skenario 2	
Accuracy training	96,5 %
Accuracy validation	95 %



Gambar 12. Visualisasi akurasi dan loss skenario 2

Dari Gambar 12 dapat diketahui bahwa hasil *training* model yang dibangun memiliki performa yang baik. Model tersebut mampu mencapai *accuracy training* sebesar 96,5% dan *accuracy validation* sebesar 95%. Hal tersebut menunjukkan bahwa model ini mampu melakukan klasifikasi gambar dengan efektif tanpa mengalami *overfitting* atau *underfitting*.

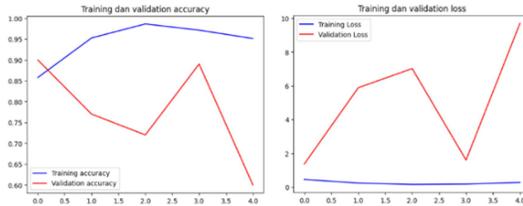
3.5.3. Skenario 3

Pada skenario ketiga, model dibangun dengan konfigurasi *unfreeze* pada *base model* serta *learning rate* sebesar 0.001 dan nilai *epoch* 5. Data yang digunakan sebanyak 200 gambar untuk *training* dan 25 gambar untuk *validation*. Pengecekan terhadap *accuracy training* dan *accuracy validation*, serta *loss training* dan *loss validation* dilakukan untuk

memonitor apakah model yang dibangun mengalami *underfitting* atau *overfitting*. Hal tersebut sangat berguna dan penting untuk mengetahui bagaimana performa model yang dihasilkan.

Tabel 4. Hasil Skenario 3

Skenario 3	
Accuracy training	95,13 %
Accuracy validation	60 %



Gambar 13. Visualisasi dan loss skenario 3

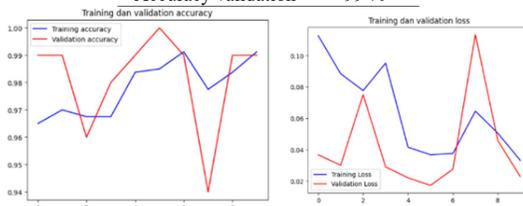
Dari Gambar 13 dapat diketahui bahwa hasil *training* model yang dibangun memiliki performa yang kurang baik. Model tersebut memiliki *accuracy training* sebesar 95,13% sedangkan *accuracy validation* sebesar 60% yang menandakan adanya indikasi bahwa model tersebut mengalami *overfitting*. Hal tersebut diperjelas pada plot grafik terhadap *epoch* diatas yang mana grafik *training accuracy* terus meningkat sedangkan grafik *validation accuracy* mengalami fluktuatif. Hal ini menunjukkan bahwa model tersebut tidak baik jika akan digunakan untuk melakukan klasifikasi gambar.

3.5.4. Skenario 4

Pada skenario keempat, model dibangun dengan konfigurasi *freeze* pada *base model* serta *learning rate* sebesar 0.001 dan nilai *epoch* 10. Data yang digunakan sebanyak 200 gambar untuk *training* dan 25 gambar untuk *validation*. Pengecekan terhadap *accuracy training* dan *accuracy validation*, serta *loss training* dan *loss validation* dilakukan untuk memonitor apakah model yang dibangun mengalami *underfitting* atau *overfitting*. Hal tersebut sangat berguna dan penting untuk mengetahui bagaimana performa model yang dihasilkan.

Tabel 5. Hasil Skenario 4

Skenario 4	
Accuracy training	99,12 %
Accuracy validation	99 %



Gambar 14. Visualisasi akurasi dan loss skenario 4

Dari Gambar 14 dapat diketahui bahwa hasil *training* model yang dibangun memiliki performa

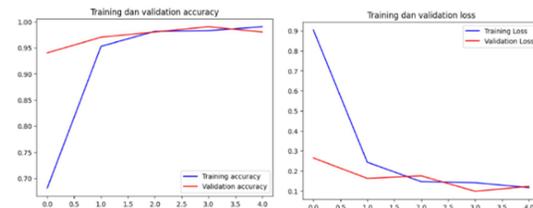
yang baik. Model tersebut mampu mencapai *accuracy training* sebesar 99,12% dan *accuracy validation* sebesar 99%. Walaupun dengan grafik yang mengalami fluktuatif, tetapi hasil dari kedua *accuracy* tidak memiliki perbedaan yang signifikan. Hal tersebut menunjukkan bahwa model ini mampu melakukan klasifikasi gambar dengan efektif tanpa mengalami *overfitting* atau *underfitting*.

3.5.5. Skenario 5

Pada skenario kelima, model dibangun dengan konfigurasi *unfreeze* pada *base model* serta *learning rate* sebesar 0.0001 dan nilai *epoch* 5. Data yang digunakan sebanyak 200 gambar untuk *training* dan 25 gambar untuk *validation*. Pengecekan terhadap *accuracy training* dan *accuracy validation*, serta *loss training* dan *loss validation* dilakukan untuk memonitor apakah model yang dibangun mengalami *underfitting* atau *overfitting*. Hal tersebut sangat berguna dan penting untuk mengetahui bagaimana performa model yang dihasilkan.

Tabel 6. Hasil Skenario 5

Skenario 5	
Accuracy training	99 %
Accuracy validation	98 %



Gambar 15. Visualisasi akurasi dan loss skenario 5

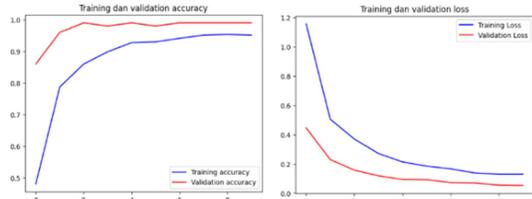
Dari Gambar 15 dapat diketahui bahwa hasil *training* model yang dibangun memiliki performa yang baik. Model tersebut mampu mencapai *accuracy training* sebesar 99% dan *accuracy validation* sebesar 98%. Hal tersebut menunjukkan bahwa model ini mampu melakukan klasifikasi gambar dengan efektif tanpa mengalami *overfitting* atau *underfitting*.

3.5.6. Skenario 6

Pada skenario keenam, model dibangun dengan konfigurasi *freeze* pada *base model* serta *learning rate* sebesar 0.0001 dan nilai *epoch* 10. Data yang digunakan sebanyak 200 gambar untuk *training* dan 25 gambar untuk *validation*. Pengecekan terhadap *accuracy training* dan *accuracy validation*, serta *loss training* dan *loss validation* dilakukan untuk memonitor apakah model yang dibangun mengalami *underfitting* atau *overfitting*. Hal tersebut sangat berguna dan penting untuk mengetahui bagaimana performa model yang dihasilkan.

Tabel 7. Hasil Skenario 6

Skenario 6	
Accuracy training	95,13 %
Accuracy validation	99 %



Gambar 16. Visualisasi akurasi dan loss skenario 6

Dari Gambar 16 dapat diketahui bahwa hasil *training* model yang dibangun memiliki performa yang baik. Model tersebut mampu mencapai *accuracy training* sebesar 95,13% dan *accuracy validation* sebesar 99%. Hal tersebut menunjukkan bahwa model ini mampu melakukan klasifikasi gambar dengan efektif tanpa mengalami *overfitting* atau *underfitting*.

3.5.7. Pengujian Data Test

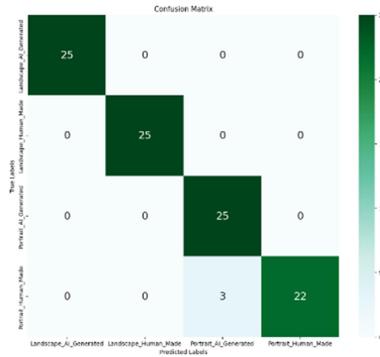
Pengujian data *test* adalah proses untuk menilai model terbaik dari 6 skenario yang telah dibangun sebelumnya. Untuk setiap skenario yang telah dibangun, akan dipilih satu model terbaik dengan nilai *accuracy test* tertinggi dan tidak memiliki nilai yang jauh berbeda dengan *accuracy validation*. Nilai *accuracy test* dan *accuracy validation* merupakan indikator bagi efektivitas sebuah model dalam menangani data yang belum pernah dilihat oleh model tersebut. Pada *tensorflow* nilai *accuracy test* dapat diuji menggunakan fungsi *evaluate* yang mengembalikan nilai *loss* dan *accuracy*. Pengujian ini dilakukan menggunakan data *test* yang memiliki total 100 gambar dengan masing-masing kelas memiliki 25 gambar. Perbandingan model terbaik pada setiap skenario dapat dilihat pada tabel 8.

Tabel 8. Perbandingan Model

Skenario	Accuracy training	Accuracy validation	Accuracy test
1	62,63 %	25 %	25 %
2	96,5 %	95 %	93 %
3	95,13 %	60 %	60 %
4	99,12 %	99 %	95 %
5	99 %	98 %	97 %
6	95,13 %	95 %	92 %

Berdasarkan perbandingan tersebut, model pada skenario 5 dapat diidentifikasi sebagai model yang terbaik untuk evaluasi. Model yang dibangun pada skenario ini menerapkan konfigurasi *unfreeze* pada *base model* serta *learning rate* sebesar 0.0001 dan nilai *epoch* 5. Metode evaluasi yang digunakan adalah *confusion matrix* berukuran 4x4, sesuai dengan 4 kelas yang ada. *Confusion matrix* ini digunakan untuk menyajikan informasi tentang kelas sebenarnya dari contoh-contoh dan hasil prediksi model. Dengan menggunakan *confusion matrix*, performa model dapat dipahami dengan jelas. Selain itu, metode ini memungkinkan perhitungan metrik seperti *accuracy*, *precision*, *recall*, dan *f1-score* yang penting untuk evaluasi kinerja model. Nilai-nilai ini memberikan gambaran yang lebih lengkap mengenai kemampuan model dalam

mengklasifikasikan data ke berbagai kelas yang berbeda. *Confusion matrix* dari hasil pengujian model pada data uji ditampilkan secara visual pada Gambar 17.



Gambar 17. Hasil confusion matrix

Selanjutnya, evaluasi dilakukan dengan menghitung *precision*, *recall*, *f1-score*, dan *accuracy* untuk semua kelas yang ada. Pendekatan ini bertujuan untuk memperoleh pemahaman yang lebih menyeluruh mengenai performa model klasifikasi dalam mengategorikan data ke berbagai kelas.

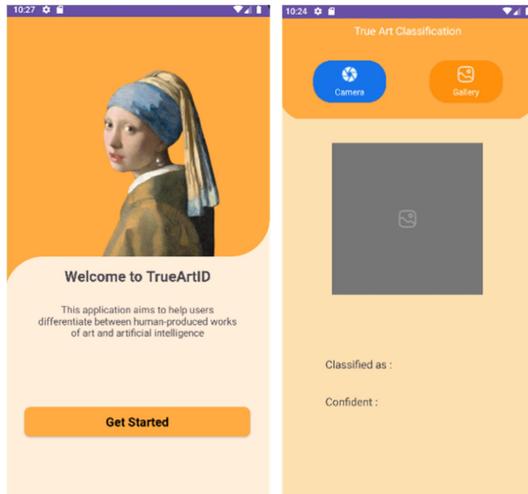
	precision	recall	f1-score	support
Landscape_AI_Generated	1.00	1.00	1.00	25
Landscape_Human_Made	1.00	1.00	1.00	25
Portrait_AI_Generated	0.89	1.00	0.94	25
Portrait_Human_Made	1.00	0.88	0.94	25
accuracy			0.97	100
macro avg	0.97	0.97	0.97	100
weighted avg	0.97	0.97	0.97	100

Gambar 18. classification report

Gambar 18 menampilkan *precision*, *recall*, dan *f1-score* untuk setiap kelas gambar yang telah dihitung dan dirata-ratakan. Rata-rata dari semua kelas menunjukkan *precision* sebesar 97%, *recall* sebesar 97%, dan *f1-score* sebesar 97%. Selain itu, *accuracy* yang diperoleh adalah 97%, dihitung dari total 100 gambar pada data *test*.

3.6. Deployment

Model terbaik yang dipilih yaitu skenario ke 5 dengan konfigurasi *unfreeze* pada *base model* serta *learning rate* sebesar 0.0001 dan nilai *epoch* 5 akan dikonversi menjadi file dengan ekstensi *tflite* agar dapat digunakan pada *platform* android dengan bahasa pemrograman kotlin. Tujuan dari tahap *deployment* ini adalah untuk mengimplementasikan model yang telah dibuat sehingga dapat digunakan oleh pengguna untuk melakukan prediksi terhadap input gambar yang diberikan melalui aplikasi android. Pada aplikasi yang dibangun, pengguna akan diberikan pilihan untuk memasukan gambar melalui penyimpanan lokal atau kamera. Gambar 19 merupakan tampilan antarmuka aplikasi yang dibangun.



Gambar 19. Tampilan antarmuka aplikasi

Setelah pengguna memilih melalui penyimpanan lokal atau kamera, selanjutnya aplikasi akan melakukan penyesuaian gambar yang diinputkan. Penyesuaian tersebut meliputi perubahan ukuran dan normalisasi agar gambar tersebut dapat diterima oleh model *tf lite*. Kemudian, model tersebut akan memberikan prediksi berdasarkan input gambar yang diberikan. Gambar 20 menampilkan aplikasi dengan model yang dibangun menggunakan arsitektur MobileNetV2 berhasil melakukan prediksi gambar hasil *generated* yang diunduh dari website Leonardo AI dengan benar dan *confidence* sebesar 89,98%.



Gambar 20. Tampilan antarmuka setelah melakukan prediksi

4. DISKUSI

Hasil penelitian ini menunjukkan bahwa arsitektur MobileNetV2 yang digunakan dalam klasifikasi gambar karya seni lukis manusia dan kecerdasan buatan mampu menghasilkan performa yang cukup baik. Model terbaik dengan konfigurasi *learning rate* 0.0001, *unfreeze* pada *base model*, dan

epoch sebanyak 5 berhasil mencapai rata-rata akurasi, *precision*, *recall*, dan *f1-score* sebesar 97% pada data uji. Model ini juga menunjukkan performa yang stabil dalam klasifikasi gambar, tanpa indikasi *overfitting* atau *underfitting*, serta memberikan prediksi yang akurat dalam aplikasi Android, dengan *confidence* sebesar 89,98%.

Jika dibandingkan dengan beberapa penelitian sebelumnya, hasil penelitian ini bersaing dengan beberapa arsitektur CNN yang lebih kompleks. Sebagai contoh, penelitian yang dilakukan oleh Johannes et al. [11] membandingkan arsitektur VGG-19 dan ResNet-50 dalam klasifikasi lukisan Van Gogh. Mereka mendapatkan akurasi sebesar 93,28% untuk VGG-19 dan 90,28% untuk ResNet-50, yang lebih rendah daripada akurasi 97% yang dicapai oleh MobileNetV2 dalam penelitian ini. Meskipun VGG-19 dan ResNet-50 merupakan model yang lebih besar, mereka tidak selalu memberikan performa yang lebih baik daripada MobileNetV2, terutama dalam skenario aplikasi mobile dengan sumber daya yang terbatas.

Penelitian lain oleh Richardson et al. [12] membandingkan berbagai arsitektur CNN, yaitu VGG-16, VGG-19, ResNet, dan Xception, dalam klasifikasi pencipta lukisan terkenal. Dalam studi ini, arsitektur Xception berhasil mencapai akurasi tertinggi sebesar 95%, diikuti oleh ResNet dengan akurasi 92%, sedangkan VGG hanya mampu mencapai akurasi sekitar 60%. Perbandingan ini menunjukkan bahwa MobileNetV2, dengan akurasi 97%, mampu bersaing bahkan dengan Xception yang dikenal memiliki arsitektur lebih kompleks, namun dengan kelebihan dari segi efisiensi komputasi yang lebih baik.

Perbandingan dengan penelitian sebelumnya juga menggarisbawahi pentingnya konfigurasi model yang tepat. Penelitian ini membuktikan bahwa kombinasi antara *unfreeze* pada *base model*, *learning rate* yang lebih kecil, dan *epoch* yang optimal dapat menghindari masalah *overfitting* dan menghasilkan model yang lebih stabil. Hal ini terlihat dari hasil pengujian yang menunjukkan konsistensi antara akurasi *training*, *validation*, dan *test*. Dengan demikian, diharapkan penelitian ini memberikan kontribusi yang signifikan dalam memahami bagaimana arsitektur CNN ringan seperti MobileNetV2 dapat dioptimalkan untuk klasifikasi gambar seni lukis, bahkan ketika dibandingkan dengan arsitektur yang lebih kompleks seperti VGG, ResNet, dan Xception. MobileNetV2 tidak hanya memberikan performa akurasi yang lebih tinggi, tetapi juga lebih efisien dalam hal penggunaan sumber daya, membuatnya sangat cocok untuk implementasi pada perangkat mobile atau Android.

5. KESIMPULAN

Penelitian ini berhasil mengembangkan model klasifikasi gambar yang efektif menggunakan arsitektur MobileNetV2 untuk membedakan lukisan

hasil karya manusia dan kecerdasan buatan. Dari enam skenario pengujian yang diterapkan, model terbaik ditemukan pada Skenario 5, yang menggunakan konfigurasi *unfreeze* pada *base model*, *learning rate* 0.0001, dan 5 *epoch*. Model ini tidak hanya mencapai akurasi tinggi sebesar 97%, tetapi juga menunjukkan stabilitas performa tanpa adanya masalah *overfitting* atau *underfitting*. Implementasi hasil penelitian ini dalam bentuk aplikasi Android juga membuktikan kapabilitas model untuk digunakan dalam lingkungan nyata. Aplikasi tersebut memungkinkan pengguna memasukkan gambar secara langsung dari penyimpanan atau kamera, dan model dapat memprediksi dengan tingkat *confidence* yang tinggi, yaitu sebesar 89,98%.

Dengan hasil ini, diharapkan penelitian ini memberikan kontribusi penting dalam penerapan teknologi *deep learning* khususnya dalam konteks klasifikasi karya seni. Arsitektur MobileNetV2 terbukti sebagai solusi yang efisien untuk digunakan pada perangkat mobile, terutama dalam kondisi sumber daya yang terbatas, tanpa mengorbankan akurasi. Penelitian ini juga membuka jalan bagi eksplorasi lebih lanjut terkait penerapan model-model ringan seperti MobileNetV2 dalam bidang klasifikasi gambar dan aplikasi praktis.

DAFTAR PUSTAKA

- [1] S. Salam, B. Sukarman, Hasnawati, and M. Muh, *Pengetahuan Dasar Seni Rupa*. 2020. [Online]. Available: [http://eprints.unm.ac.id/18130/1/Pengetahuan Dasar Seni Rupa 2020 lengkap.pdf](http://eprints.unm.ac.id/18130/1/Pengetahuan%20Dasar%20Seni%20Rupa%20lengkap.pdf)
- [2] N. I. Rahmahafida and W. B. Sinag, "Analisis Problematika Lukisan Ciptaan Artificial Intelligence Menurut Undang-Undang Hak Cipta," *J. Pendidik. dan Konseling*, vol. 4, no. 6, pp. 9688–9696, 2022.
- [3] A. Xue, "End-to-end Chinese landscape painting creation using generative adversarial networks," *Proc. - 2021 IEEE Winter Conf. Appl. Comput. Vision, WACV 2021*, pp. 3862–3870, 2021, doi: 10.1109/WACV48630.2021.00391.
- [4] M. Ragot, N. Martin, and S. Cojean, "AI-generated vs. human artworks. a perception bias towards artificial intelligence?," *Conf. Hum. Factors Comput. Syst. - Proc.*, 2020, doi: 10.1145/3334480.3382892.
- [5] Y. Tian, "Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm," *IEEE Access*, vol. 8, pp. 125731–125744, 2020, doi: 10.1109/ACCESS.2020.3006097.
- [6] T. A. Bowo, H. Syaputra, and M. Akbar, "Penerapan Algoritma Convolutional Neural Network Untuk Klasifikasi Motif Citra Batik Solo," *J. Softw. Eng. Ampera*, vol. 1, no. 2, pp. 82–96, 2020, doi: 10.51519/journalsea.v1i2.47.
- [7] Q. N. Azizah, "Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network AlexNet," *sudo J. Tek. Inform.*, vol. 2, no. 1, pp. 28–33, 2023, doi: 10.56211/sudo.v2i1.227.
- [8] R. Holiyanti, S. Wati, I. Fahmi, and C. Rozikin, "Pendeteksi Sampah Metal untuk Daur Ulang Menggunakan Metode Convolutional Neural Network," *J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 1, pp. 261–274, 2022, doi: 10.28932/jutisi.v8i1.4492.
- [9] M. K. Soegeng, L. Liliana, and A. Noertjahyana, "Penerapan Convolutional Neural Network untuk Klasifikasi Kanker Kulit Melanoma pada Dataset Gambar Kulit," *J. Infra*, vol. 9, no. 1, pp. 47–51, 2021, [Online]. Available: <https://publication.petra.ac.id/index.php/teknik-informatika/article/view/10945>
- [10] F. N. Cahya, N. Hardi, D. Riana, and S. Hadiyanti, "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)," *Sistemasi*, vol. 10, no. 3, p. 618, 2021, doi: 10.32520/stmsi.v10i3.1248.
- [11] Y. Yohannes, D. Udjulawa, and F. Febbiola, "Klasifikasi Lukisan Karya Van Gogh Menggunakan Convolutional Neural Network-Support Vector Machine," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 192–205, 2021, doi: 10.28932/jutisi.v7i1.3399.
- [12] R. S. T. Menezes, A. M. Cordeiro, R. Magalhães, and H. Maia, "Classification of Paintings Authorship Using Convolutional Neural Network," no. October, pp. 1–8, 2021, doi: 10.21528/cbic2021-116.
- [13] K. Mondal, "Categorization of artwork images based on painters using CNN," *J. Phys. Conf. Ser.*, vol. 1818, no. 1, 2021, doi: 10.1088/1742-6596/1818/1/012223.
- [14] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 Model for Image Classification," *Proc. - 2020 2nd Int. Conf. Inf. Technol. Comput. Appl. ITCA 2020*, pp. 476–480, 2020, doi: 10.1109/ITCA52113.2020.00106.
- [15] R. Indraswari, R. Rokhana, and W. Herulambang, "Melanoma image classification based on MobileNetV2 network," *Procedia Comput. Sci.*, vol. 197, pp. 198–207, 2021, doi: 10.1016/j.procs.2021.12.132.
- [16] Statcounter GlobalStats, "Mobile Operating System Market Share Indonesia," 2023. <https://gs.statcounter.com/os-market-share/mobile/indonesia>

- [17] C. E. D. Vanegas, J. C. G. Mejía, F. A. V. Agudelo, and D. E. S. Duran, "A Representation Based on Essence for the CRISP-DM Methodology," *Comput. y Sist.*, vol. 27, no. 3, pp. 675–689, 2023, doi: 10.13053/CyS-27-3-3446.
- [18] J. A. Solano, D. J. Lancheros Cuesta, S. F. Umaña Ibáñez, and J. R. Coronado-Hernández, "Predictive models assessment based on CRISP-DM methodology for students performance in Colombia - Saber 11 Test," *Procedia Comput. Sci.*, vol. 198, no. 2020, pp. 512–517, 2021, doi: 10.1016/j.procs.2021.12.278.
- [19] W. Y. Ayele, "Adapting CRISP-DM for idea mining a data mining process for generating ideas using a textual dataset," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 6, pp. 20–32, 2020, doi: 10.14569/IJACSA.2020.0110603.
- [20] A. B. J. Malyan, R. R. Putra, E. Laila, A. T. Wardhana, M. Fikri, and I. G. T. Isa, *Epoch Tuning Hyperparameter in Fire Image Classification at University Sjakhyakirti*, no. July. Atlantis Press International BV, 2023. doi: 10.2991/978-94-6463-118-0_37.
- [21] S. Khasim, H. Ghosh, I. S. Rahat, K. Shaik, and M. Yesubabu, "Deciphering Microorganisms through Intelligent Image Recognition: Machine Learning and Deep Learning Approaches, Challenges, and Advancements," *EAI Endorsed Trans. Internet Things*, vol. 10, pp. 1–8, 2024, doi: 10.4108/eetiot.4484.
- [22] Z. Geng, Y. Xu, B. N. Wang, X. Yu, D. Y. Zhu, and G. Zhang, "Target Recognition in SAR Images by Deep Learning with Training Data Augmentation," *Sensors*, vol. 23, no. 2, 2023, doi: 10.3390/s23020941.

