



PENGEMBANGAN SISTEM DETEKSI MEDIA BUATAN AI MENGGUNAKAN ARSITEKTUR CNN *RESNET-50*

Guruh Pratama Putra¹, Adam Sekti Aji²

^{1,2} Informatika, Universitas Teknologi Yogyakarta
 Yogyakarta, D.I Yogyakarta, Indonesia 55164
guruhputra1412@gmail.com, adamaji@uty.ac.id

Abstract

The rapid advancement of artificial intelligence (AI) technology has led to an increase in AI-generated media that is increasingly difficult to distinguish from authentic media. This phenomenon presents significant challenges in various fields, including digital security and the creative industry, necessitating a reliable automatic detection system. Manual identification is inefficient and error-prone, as it requires specialized expertise to recognize subtle digital media. This study proposes the development of an AI-generated media detection system using a Convolutional Neural Network (CNN) with the ResNet-50 architecture. The ResNet-50 model was chosen due to its proven ability to handle deep feature extraction and overcome the vanishing gradient problem. A sample of 4.600 images, 2.300 AI images and 2.300 real images, were used. The research methodology included data collection from various Kaggle datasets, data preprocessing including resizing and augmentation, model training, and performance evaluation based on accuracy, precision, and recall metrics. Experimental results show that the developed model achieved a high testing accuracy of 97.30%. This indicates the model's capability to effectively classify media as AI-generated or real with high precision. This research is expected to be a valuable reference for the development of more accurate and efficient AI systems in detecting synthetic media.

Keywords: AI Media Detection, CNN, Deep Learning, ResNet-50, Transfer Learning

Abstrak

Kemajuan pesat teknologi kecerdasan buatan (AI) telah menyebabkan peningkatan media buatan AI yang semakin sulit dibedakan dari media autentik. Fenomena ini menghadirkan tantangan signifikan di berbagai bidang, termasuk keamanan digital dan industri kreatif, sehingga memerlukan sistem deteksi otomatis yang andal. Identifikasi manual tidak efisien dan rentan terhadap kesalahan, karena membutuhkan keahlian khusus untuk mengenali media digital yang halus. Penelitian ini mengusulkan pengembangan sistem deteksi media buatan AI menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur *ResNet-50*. Model *ResNet-50* dipilih karena kemampuannya yang telah terbukti dalam menangani ekstraksi fitur mendalam dan mengatasi masalah *vanishing gradient*. Terdapat 4.600 gambar dengan 2.300 gambar AI dan 2.300 gambar asli. Metodologi penelitian meliputi pengumpulan data dari berbagai *dataset* Kaggle, *preprocessing* data termasuk perubahan ukuran dan augmentasi, pelatihan model, serta evaluasi performa berdasarkan akurasi, *presicion*, dan *recall*. Hasil eksperimen menunjukkan bahwa model yang dikembangkan mencapai akurasi pengujian yang tinggi sebesar 97,30%. Hal ini menandakan kapabilitas model untuk secara efektif mengklasifikasikan media sebagai buatan AI atau asli dengan presisi tinggi. Penelitian ini diharapkan dapat menjadi referensi berharga untuk pengembangan sistem AI yang lebih akurat dan efisien dalam mendeteksi media sintesis.

Kata kunci: CNN, Deteksi Media AI, Deep Learning, ResNet-50, Transfer Learning

1. PENDAHULUAN

Kemajuan pesat dalam teknologi kecerdasan buatan atau *Artificial Intelligent (AI)* telah memicu transformasi di berbagai sektor, termasuk dalam ranah pengenalan gambar. Salah-satu tolak ukur fundamental dalam pengembangan AI adalah pencapaian kemampuan untuk menginterpretasi dan mengategorikan data visual secara akurat. Untuk tugas

spesifik seperti klasifikasi gambar, *Convolutional Neural Network* (CNN) telah muncul sebagai salah-satu metodologi yang paling efektif dan berhasil hingga saat ini[1]. Gambar buatan AI ini sedang ramai diperbincangkan di media sosial sekarang. Gambar tersebut dibuat dengan cara memasukkan deskripsi perintah, gambar seperti apa yang pengguna inginkan. Salah -satu model AI terkenal untuk membuat

gambar adalah *Stable Diffusion*. Dengan maraknya gambar buatan AI ini, orang-orang mulai khawatir tidak bisa membedakan gambar buatan AI dengan gambar asli yang dibuat oleh tangan manusia. Arsitektur CNN didasarkan pada prinsip kerja sistem visual manusia. Sebagaimana manusia memproses informasi visual dengan mengidentifikasi pola dan fitur khas objek yang dilihat dengan pemahaman itulah CNN dirancang secara spesifik untuk meniru kemampuan tersebut guna memahami fitur yang ada pada gambar[2].

CNN merupakan salah-satu metode yang paling umum digunakan dan telah terbukti efektif dalam pemrosesan data citra untuk pengenalan pola. Pada penelitian ini, menggunakan CNN *ResNet-50*, bertujuan untuk membangun sistem yang dapat mendeteksi fitur-fitur pada gambar AI secara lebih optimal dengan mencari nilai *precision*, *recall*, dan akurasi terbaik. CNN memiliki kemampuan yang baik dalam menganalisis gambar dan mengenali pola, telah terbukti efektif dalam berbagai tugas pengolahan citra, termasuk mendeteksi gambar AI. Fokus utama penelitian ini adalah memahami bagaimana algoritma CNN dapat digunakan untuk membuat sistem deteksi media gambar dengan tujuan meningkatkan akurasi serta kecepatan dalam proses pengenalan.

Terdapat penelitian terdahulu yang relevan dengan penelitian ini, dengan memanfaatkan arsitektur *EfficientNetB0* dan integrasi *OpenCV* untuk ekstraksi *landmark* wajah, yang berhasil mencapai akurasi sebesar 91% dalam mendeteksi citra *DeepFake*[3]. Pada penelitian kali ini, penulis ingin menerapkan arsitektur *ResNet-50* yang memiliki lapisan lebih dalam guna mengoptimalkan ekstraksi fitur pada media generatif. Selain itu, menambahkan fitur deteksi video AI.

Oleh karena itu, penelitian ini akan membahas penerapan model CNN dalam mendeteksi media gambar AI. Dengan melatih jaringan serta memanfaatkan teknik ekstraksi fitur yang lebih maju, penelitian ini bertujuan untuk mengoptimalkan kemampuan model CNN dalam mengenali dan mengelompokkan gambar buatan AI secara otomatis. Pada penelitian ini juga terdapat GUI untuk menguji secara langsung media gambar dan video yang bisa diunggah.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

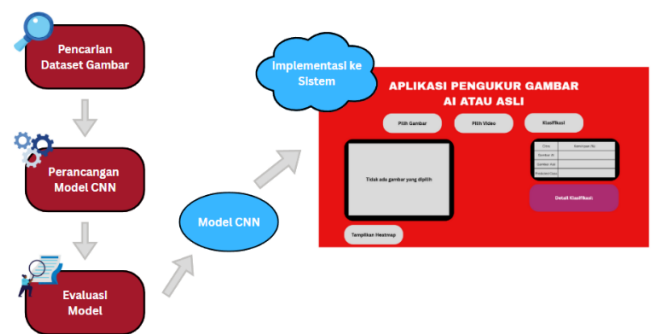
Proses pengembangan sistem deteksi media buatan AI ini dilakukan secara sistematis melalui empat tahapan utama, mulai dari perolehan data hingga implementasi ke dalam sistem antarmuka pengguna. Alur tahapan tersebut dapat dilihat pada Gambar 1. Berikut penjelasan setiap tahapan:

1) Pencarian *Dataset* Gambar

Tahap awal penelitian berfokus pada pengumpulan data sebagai bahan utama pelatihan model. Data yang digunakan

bersumber dari platform *Kaggle* dan beberapa tangkapan gambar secara mandiri. *Dataset* ini terdiri dari:

- Data *Training*:** Sebanyak 4.600 gambar yang terbagi seimbang antara 2.300 gambar asli dan 2.300 gambar buatan AI.
- Data *Testing*:** Sebanyak 1.000 gambar yang juga dibagi secara merata untuk mengukur performa akhir model.
- Data Tambahan:** Untuk menguji generalisasi sistem di luar *dataset Kaggle*, penulis juga mengumpulkan 10 gambar dari Pinterest dan 5 video buatan AI dari YouTube.



Gambar 1. Tahapan Penelitian

2) Perancangan Model CNN

Pada tahap ini, dilakukan pengembangan arsitektur model menggunakan CNN dengan fokus pada arsitektur *ResNet-50*. Perancangan model ini meliputi:

- Penerapan *Transfer Learning*:** Menggunakan model *pre-trained* yang telah dilatih pada *database ImageNet* untuk memperkuat ekstraksi fitur visual.
- Implementasi Perangkat Lunak:** Model dibangun menggunakan *library* PyTorch dengan bahasa pemrograman Python.
- Optimasi *Hyperparameter*:** Melakukan eksperimen terhadap berbagai parameter seperti *optimizer* (Adam, AdamW, dan SGD), *learning rate*, dan jumlah *epoch* untuk menemukan konfigurasi paling optimal.

3) Evaluasi Model

Tahap evaluasi bertujuan untuk mengukur efektivitas dan tingkat keandalan model yang telah dirancang. Pengujian dilakukan menggunakan metrik evaluasi seperti *Accuracy*, *Precision*, dan *Recall*. Model diuji menggunakan *data test* untuk melihat sejauh mana kemampuannya dalam menggeneralisasi data baru. Kriteria keberhasilan pelatihan ditetapkan pada ambang batas akurasi uji di atas 90%.

4) Implementasi ke Sistem

Tahap akhir adalah mengintegrasikan model yang paling optimal ke dalam aplikasi perangkat lunak berbasis *Graphical User Interface* (GUI).

- Pembuatan Antarmuka: GUI dirancang menggunakan *library* PyQt5 dan alat bantu *Qt Designer* untuk memudahkan interaksi pengguna.
- Fitur Sistem: Aplikasi ini memungkinkan pengguna untuk mengunggah media gambar (format jpg, png, jpeg, bmp) serta video (format mp4, avi, mov, mkv) untuk diklasifikasi secara otomatis.
- Analisis Visual: Hasil klasifikasi akan ditampilkan dalam sebuah tabel yang isinya berupa persentase seberapa asli atau AI gambar atau video yang diunggah.

2.2 Tinjauan Pustaka

Tinjauan pustaka memberikan gambaran penelitian-penelitian sebelumnya yang relevan dengan topik pengembangan sistem pengenalan gambar AI atau bukan dengan menggunakan algoritma CNN. Berikut merupakan jurnal penelitian yang dijadikan tinjauan pustaka.

Penelitian pertama yang penulis kaji, memiliki kesamaan dengan penelitian dari aspek metode yang digunakan dan kajian utamanya yaitu klasifikasi gambar. Metode CNN dalam penelitian ini digunakan untuk mengklasifikasikan gambar-gambar pemandangan alam secara otomatis. *Dataset* gambar tersebut dikategorikan ke dalam lima kelas, yaitu gunung, hutan, jalan, laut, dan gletser. Hasil *train* model adalah 95% berdasarkan grafik yang ditunjukkan[4].

Penelitian kedua yang penulis kaji, memiliki kesamaan dengan penelitian dari aspek metode yang digunakan dan kajian utamanya yaitu klasifikasi gambar. CNN diimplementasikan untuk tugas klasifikasi citra motif batik Yogyakarta. Dalam penelitian ini, digunakan total 600 citra yang mencakup tiga motif berbeda, yakni Ceplok, Kawung, dan Parang. Metode CNN berhasil mencapai tingkat akurasi tertinggi sebesar 87,83%. Kinerja model ini juga didukung oleh nilai *precision* rata-rata 88,46% dan nilai *recall* rata-rata 87,66%[5].

Penelitian ketiga yang penulis kaji, memiliki kesamaan dengan penelitian dari aspek metode yang digunakan dan kajian utamanya yaitu klasifikasi gambar. Teknologi CNN dimanfaatkan untuk mengklasifikasikan 400 citra batik tanah liat Sumatera Barat yang terbagi dalam 4 kategori. Dalam penelitian ini, pembagian data dilakukan dengan menetapkan 320 citra sebagai data latih dan 80 citra sisanya sebagai data uji. Evaluasi model menunjukkan nilai akurasi yang sangat tinggi pada tahap pelatihan, yakni 98,75%. Namun, terjadi penurunan akurasi yang signifikan pada tahap pengujian, yang hanya mencapai 62,5%[6].

Penelitian keempat yang penulis kaji, memiliki kesamaan dengan penelitian dari aspek metode yang digunakan dan arsitektur metode. Menggunakan metode CNN *ResNet-50* untuk mengidentifikasi varian tanaman anggur berdasarkan citra daun anggur. Data citra daun anggur yang digunakan berjumlah 150 dengan 3 varian anggur yang terdiri 50 daun Nizina, 50 daun Jupiter dan 50 daun Isabella. Data citra daun anggur diubah ukurannya dalam piksel, menghapus *background*, augmentasi data. Sistem menghasilkan tingkat akurasi *training* 86 % sedangkan akurasi validasi terbaik adalah 91%[7].

Penelitian kelima yang penulis kaji, memiliki kesamaan dengan penelitian dari aspek metode yang digunakan. Mengidentifikasi citra cuaca dengan memakai algoritma CNN. Cuaca itu mencakup *Cloudy*, *Rain*, *Sunrise*, atau *Shine*. Dalam penelitian ini, data citra dibagi menjadi 90 data latih dan 10 *data testing*. Proses pelatihan dijalankan dengan konfigurasi 150 *epoch*. Pengaturan ini menghasilkan akurasi tertinggi pada data latih sebesar 94,16%, namun akurasi pada *data testing* tercatat hanya 65,00%[8].

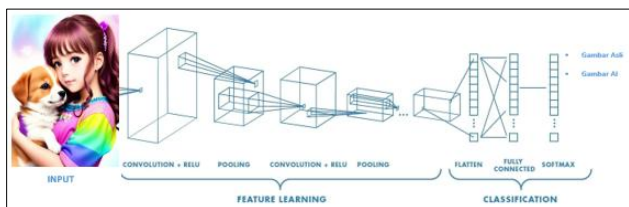
Penelitian terakhir yang penulis kaji, memiliki kesamaan dengan penelitian dari aspek metode yang digunakan dan arsitektur metode. Klasifikasi jenis kulit wajah menggunakan metode CNN dengan arsitektur *ResNet-50*. Menggunakan *dataset* gambar kulit wajah dengan 3 kelas yaitu normal, kering, dan berminyak. Hasil penelitian menunjukkan akurasi 0.9986, *loss* 0.0040, dan *F1-Score* yang tinggi. Namun, distribusi data yang tidak seimbang menyebabkan *overfitting*[9].

Penelitian kali ini akan berfokus pada mendeteksi gambar AI menggunakan CNN *ResNet-50*. Pendekatan yang digunakan adalah *end-to-end learning*, model secara otomatis mempelajari fitur-fitur yang paling relevan pada gambar, seperti tekstur, warna, dan bentuk tanpa proses ekstraksi manual.

2.3 Landasan Teori

1) Model Algoritma CNN

CNN (*Convolutional Neural Network*) merupakan sebuah metode klasifikasi citra yang memiliki keunggulan dalam mempelajari fitur-fitur esensial dari citra kompleks secara otomatis. Kinerja arsitektur CNN dalam melakukan pengenalan citra ditentukan oleh model yang diaplikasikan, yang secara garis besar terdiri dari dua tahapan utama: ekstraksi ciri (*feature extraction*) dan klasifikasi (*classification*)[5]. Pada Gambar 2 menunjukkan arsitektur CNN secara umum.



Gambar 2. Arsitektur CNN

CNN, singkatan dari *Convolutional Neural Network*, merupakan jenis Jaringan Saraf Tiruan (ANN) yang khusus dirancang untuk memproses data visual. CNN memiliki ciri khas yaitu penggunaan layer konvolusi yang mampu menangkap pola dan fitur penting dari gambar. Meskipun fokus utama CNN adalah layer konvolusi, CNN juga dapat menggabungkan jenis layer lain seperti *pooling* dan *fully connected layer* untuk membangun CNN yang lebih kompleks, yang dikenal sebagai *deep CNN*[10].

Arsitektur CNN secara fundamental terdiri dari dua tahapan utama yaitu, *feature learning* dan *classification layer*. Tahap *feature learning* berfungsi sebagai penerima dan pengolahan data masukan berupa gambar. Bagian ini tersusun atas kombinasi *convolutional layer* dan *pooling layer*. Proses yang terjadi pada lapisan-lapisan tersebut akan menghasilkan *feature maps*, yang esensinya adalah representasi numerik dari gambar. Kumpulan *feature maps* inilah yang kemudian diteruskan sebagai masukan bagi *classification layer* untuk diproses lebih lanjut[6].

Lapisan yang berisi *neuron* yang terkoneksi secara penuh (*fully connected*) dengan lapisan lain dan nantinya akan mendapat *input* dari *output layer* bagian *feature learning* yang akan diproses pada *flatten* dengan tambahan lapisan tersembunyi (*hidden*) pada *fully connected* hingga menghasilkan *output* berupa akurasi klasifikasi dari setiap kelas itulah *classification layer*[6]. Bagian dari lapisan arsitektur CNN dijelaskan pada Tabel 1.

Tabel 1. Lapisan Arsitektur CNN

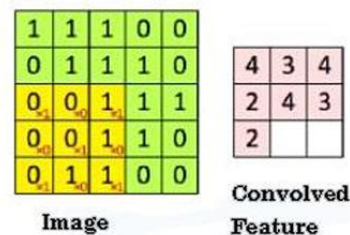
Feature Learning	a.	Input Layer
	b.	Convolution Layer
	c.	Activation Layer
	d.	Pooling Layer
Classification	e.	Fully Connected Layer
	f.	Output layer

a) Input Layer

Lapisan ini berfungsi sebagai penerima nilai-nilai piksel dari citra yang dijadikan masukan. Citra masukan tersebut terdiri dari tiga kanal warna, yakni *Red*, *Green*, *Blue* (RGB).

b) Convolution Layer

Convolutional layer terdiri dari *neuron-neuron* yang diorganisasi sedemikian rupa hingga membentuk sebuah filter dengan dimensi panjang dan tinggi (piksel) tertentu. Proses konvolusi itu sendiri adalah operasi matematis yang mengombinasikan dua matriks berbeda untuk menghasilkan sebuah matriks baru. Dalam konteks pengolahan citra, proses ini berarti mengaplikasikan sebuah *kernel* (yang esensinya adalah filter tersebut) pada sebuah citra. Sebagaimana diilustrasikan pada Gambar 3, *kernel* (kotak kuning) digeser melintasi citra pada semua *offset* yang memungkinkan[6].



Gambar 3. Gambaran Operasi Konvolusi

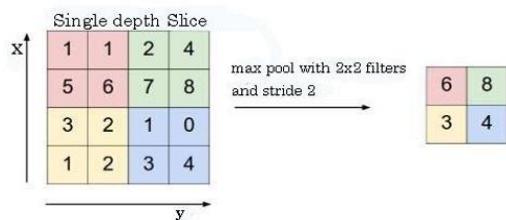
Kotak hijau pada Gambar 3 adalah representasi citra masukan yang akan diproses. Proses ekstraksi fitur dilakukan dengan menggeser *kernel* secara sistematis, yakni dari sudut kiri atas gambar menuju ke kanan bawah. Gambar di sebelah kanan (hasil konvolusi) menunjukkan *feature map* yang dihasilkan dari operasi tersebut.

c) Activation Layer

Pada lapisan ini, setiap nilai (*value*) di dalam *feature map* akan dimodifikasi dengan menerapkan fungsi aktivasi. Fungsi yang digunakan dalam konteks ini adalah ReLU (*Rectified Linear Unit*). Pada dasarnya, ReLU bekerja dengan menerapkan pembatasan nilai (*threshold*) dari 0 hingga tak terhingga. Secara praktis, ini berarti setiap nilai negatif dalam *feature map* akan diubah menjadi 0, sementara semua nilai positif akan dipertahankan apa adanya. Fungsi ini telah menjadi salah-satu fungsi aktivasi yang paling populer digunakan saat ini[6].

d) Pooling Layer

Proses *pooling* diaplikasikan pada setiap *feature map* untuk mengurangi ukurannya, yang secara efektif menurunkan resolusi spasial. Metode yang lazim digunakan adalah filter 2x2 dengan *stride* 2, yang beroperasi pada setiap irisan masukan secara terpisah[6]. Berikut ini Gambar 4 merupakan contoh gambar operasi *max-pooling*:

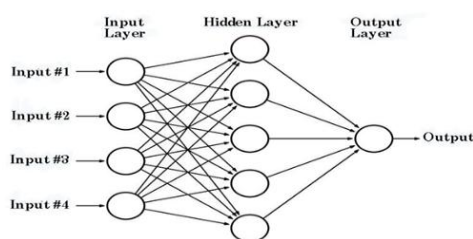


Gambar 4. Max-pooling

Kotak warna-warni yang di sebelah kiri menunjukkan masing-masing kelompok berdasarkan warna. Lalu kelompok tersebut diambil nilai maksimumnya. Sehingga hasilnya bisa dilihat pada kumpulan kotak yang di sebelah kanan.

e) Fully Connected Layer

Lapisan *Fully-Connected* (FC) atau *dense layer* mengadopsi struktur yang identik dengan lapisan pada Jaringan Saraf Tiruan (JST) standar. Maksudnya, setiap *neuron* pada lapisan ini akan menerima masukan dari semua *neuron* (atau keluaran) yang berasal dari lapisan sebelumnya[6]. Gambar 5 merupakan ilustrasi dari *fully connected layer*.



Gambar 5. Fully Connected Layer

f) Output Layer

Pada tahap akhir, yakni klasifikasi, keluaran dari lapisan sebelumnya akan diproses menggunakan fungsi aktivasi *softmax*. Fungsi tersebut akan menghitung skor probabilitas untuk setiap kelas target. Berdasarkan skor probabilitas ini, masukan akan dikategorikan ke dalam salah satu dari dua label: "Gambar AI" atau "Gambar Asli".

Keunggulan CNN dalam mengekstraksi fitur-fitur data secara efektif bersumber dari pemanfaatan lapisan konvolusi. Kemampuan fundamental inilah yang menjadikannya sebuah metodologi yang sangat andal untuk tugas-tugas seperti pengenalan pola pada citra[11].

2) Artificial Intelligence (AI)

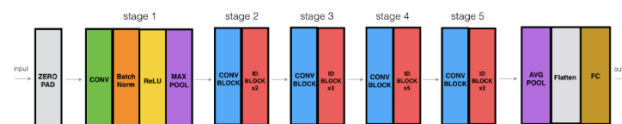
Artificial Intelligence (AI) dapat didefinisikan sebagai teknologi dalam sistem komputer yang dirancang untuk menjalankan tugas-tugas yang menyerupai kapabilitas intelektual manusia. Ciri khas utama dari AI adalah kemampuannya untuk beroperasi secara otomatis dan beradaptasi melalui proses belajar. Sistem AI menganalisis data yang diterima untuk memahami dan mengatasi berbagai masalah. Proses pembelajaran ini memungkinkannya membangun "pengetahuan", sehingga

kapabilitas program AI bersifat dinamis dan akan terus meningkat seiring dengan bertambahnya jumlah data yang diolah[12].

Jenis-jenis Kecerdasan Buatan:

- 1) Kecerdasan Buatan Lemah (*Weak AI*) – Sistem yang dirancang untuk meniru perilaku cerdas tanpa benar-benar memiliki pemahaman atau kesadaran seperti manusia.
- 2) Kecerdasan Buatan Kuat (*Strong AI*) – Sistem yang mampu berpikir dan beroperasi layaknya manusia, dengan kemampuan memahami, bernalar, serta mengambil keputusan secara mandiri.
- 3) *ResNet-50*

ResNet-50 adalah model kecerdasan buatan canggih yang dirancang untuk mengklasifikasikan gambar. Model ini memiliki 50 lapis yang telah dilatih pada lebih dari 1 juta gambar dari *database ImageNet*, membuatnya sangat ahli dalam mengenali berbagai objek dan kategori[9]. Berikut merupakan arsitektur *ResNet-50* dapat dilihat pada Gambar 6[13].

Gambar 6. Arsitektur *ResNet-50*

ResNet menawarkan koneksi instan identitas yang melewati banyak lapisan dan menggunakan fungsi aktivasi dari lapisan sebelumnya. Gagasan seperti itu memungkinkan kita merancang DNN buatan dengan banyak lapisan dan tidak khawatir tentang masalah *gradient* yang hilang. Untuk itu, arsitektur modular dikembangkan yang menumpuk *blok residu* dengan pemetaan identitas[7]. Persamaan 4 dihitung di *blok* yang tersisa:

$$x_l + 1 = x_l + (x_l \cdot W_l)$$

x_l adalah masukan dan keluaran dari *blok* sisa ke- l . W_l adalah himpunan bobot, dan $F(x_l \cdot W_l)$ adalah fungsi sisa. Persamaan 5 dapat diterapkan dengan "koneksi langsung" dengan pemetaan identitas dan melewati satu atau lebih lapisan[7]. Masalah ini diulang untuk setiap *blok* L yang tersisa untuk mendapatkan:

$$x_L = x_l + \sum_{i=1}^{L-1} (x_i \cdot W_i)$$

Dari *gradien* fungsi *loss*, lapisan *gradient* tidak hilang jika bobotnya sangat kecil. Namun oleh karena itu, masih belum ada informasi tentang letak bagian-bagian objek tersebut[7].

4) Deep Learning

Deep Learning sebuah sub-bidang dari Kecerdasan Buatan, telah membawa perubahan transformatif pada ranah pengenalan citra digital. Metodologi ini didasarkan pada

inspirasi dari cara kerja otak manusia[14]. Dengan memanfaatkan arsitektur Jaringan Saraf Tiruan yang berlapis-lapis dan kompleks, *Deep Learning* memberikan kapabilitas pada komputer untuk belajar secara langsung dari data mentah, seperti gambar. Arsitektur inilah yang memungkinkan komputer untuk "melihat" dan menginterpretasi dunia visual layaknya manusia, memberinya kemampuan untuk mengenali objek serta mendeteksi benda-benda di sekitarnya[15].

5) *Transfer Learning*

Transfer learning adalah sebuah teknik dalam *machine learning* ketika sebuah model yang telah dikembangkan untuk suatu tugas, digunakan kembali sebagai titik awal untuk model pada tugas kedua yang berbeda namun terkait [16].

Dalam klasifikasi citra dengan CNN, *transfer learning* biasanya melibatkan penggunaan model pra-latih (*pretrained model*). Model ini, seperti *ResNet-50*, telah dilatih pada *dataset* yang sangat besar (contohnya *ImageNet*) dan mampu mengidentifikasi jutaan gambar dari seribu kategori. Oleh karena itu, mereka memiliki kemampuan ekstraksi fitur visual yang sangat kuat.

Proses penerapan *transfer learning* dalam penelitian ini adalah sebagai berikut:

- Menggunakan *Backbone Pretrained* : Arsitektur *ResNet-50* yang sudah dilatih di *ImageNet* digunakan sebagai pengekstraksi fitur utama. Lapisan awal hingga menengah pada model ini sangat baik dalam mengenali fitur-fitur universal seperti tepi, tekstur, dan bentuk.
- Mengganti Lapisan Klasifikasi : Lapisan terakhir atau *fully connected layer* dari model *ResNet-50* dibuang dan diganti dengan lapisan klasifikasi baru yang disesuaikan dengan jumlah kelas pada penelitian ini, yaitu 2 kelas gambar, gambar asli dan gambar AI.
- Fine-Tuning* (Pelatihan Ulang) : Model dengan lapisan *fully connected layer* baru tersebut kemudian dilatih kembali (*fine-tuned*) menggunakan *dataset* gambar yang sudah disiapkan. Proses ini memungkinkan model untuk menyesuaikan bobotnya agar dapat mengenali fitur-fitur yang spesifik dari gambar AI.

Keuntungan dari menggunakan *Transfer Learning* adalah membuat proses pelatihan lebih cepat dan meningkatkan hasil akurasi pelatihan.

6) GUI (*Graphical User Interface*)

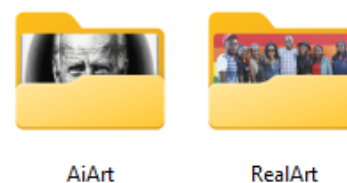
GUI (*Graphical User Interface*) adalah antarmuka pengguna yang menggunakan elemen visual seperti *window*, tombol, menu, dan dialog untuk memfasilitasi interaksi antara pengguna dan perangkat lunak atau sistem komputer. Melalui GUI, seorang pengguna dapat mengoperasikan sebuah perangkat lunak. Pengguna

memberikan perintah menggunakan perangkat keras *input*, misalnya *mouse* dan *keyboard*, dan kemudian menerima umpan balik dari sistem dalam bentuk tampilan visual atau grafis[8].

Pada penelitian ini, pembuatan GUI dibantu dengan *library* Python yaitu PyQt5. Karena dengan menggunakan *library* tersebut akan sangat memudahkan dalam membuat GUI. PyQt5 memiliki aplikasi *designer* yaitu *Qt designer*, hal itulah yang membantu memudahkan dalam mendesain GUI.

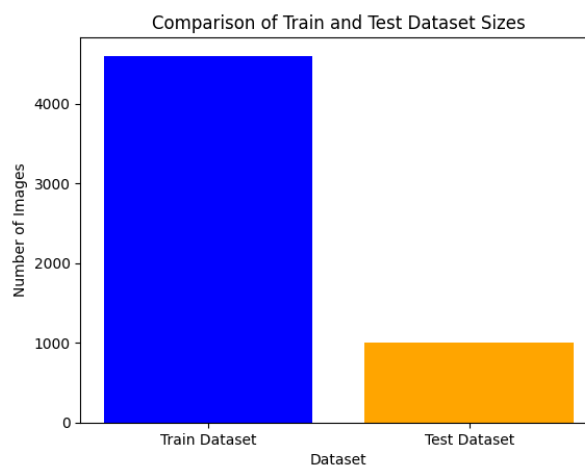
2.4 Sumber Data

Data yang digunakan dalam penelitian ini diperoleh dari sebuah *dataset* yang tersedia di platform Kaggle, yang dikenal sebagai tempat penyedia berbagai *dataset* untuk keperluan analisis data. *Dataset* tersebut berisi gambar-gambar buatan AI dan asli, yang sudah dibagi menjadi dua folder. Nama foldernya akan disesuaikan dengan isi gambar. Jadi, memungkinkan *dataset* digunakan untuk pelatihan dan pengujian model klasifikasi dengan algoritma CNN. Pada Gambar 7 menunjukkan gambar folder *dataset* yang sudah dipisah berdasarkan isi gambar.



Gambar 7. Folder *Dataset*

Jumlah gambar di *dataset training data* adalah 4.600, sedangkan jumlah gambar di *dataset test data* adalah 1000. Untuk pembagian dari gambar AI dan gambar asli dibuat sama jumlahnya di setiap *dataset*. Jumlah gambar di *dataset training data* dibuat lebih banyak agar model belajar dengan baik. Berikut perbandingan *training data* dengan *test data* pada Gambar 8:

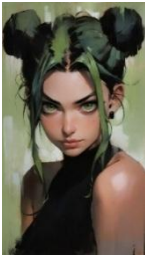
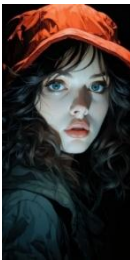


Gambar 8. Perbandingan *Train* dan *Test Dataset*

Selain gambar yang ada di *dataset train* dan *test*, penulis sudah mengumpulkan 10 gambar yang ada pada *website*

Pinterest yang nantinya digunakan untuk menguji kinerja aplikasi. Gambar-gambar uji coba bisa dilihat pada Tabel 2.

Tabel 2. Gambar Uji Coba Aplikasi

Kelas	Gambar
AI	
	
AI	
	
AI	
	
Asli	

Kelas	Gambar
Asli	
Asli	
Asli	
Asli	

Selain gambar yang ada di dataset train dan test, penulis sudah mengumpulkan 10 gambar yang ada pada website Pinterest yang nantinya digunakan untuk menguji kinerja aplikasi. Seberapa baik aplikasi yang dibuat ketika menguji gambar yang bukan dari dataset. Dari 10 gambar tersebut, terbagi menjadi 5 gambar AI dan 5 gambar asli. Gambar-gambar uji coba bisa dilihat pada Tabel 2. Penulis juga sudah mengumpulkan 5 video AI yang ada di platform YouTube untuk diuji coba pada sistem yang telah dibuat. Untuk memverifikasi video yang diuji itu AI, sang pemilik video sudah mencantumkan di judul video, di deskripsi, atau di kanalnya langsung. Video-video uji coba bisa dilihat pada Tabel 3.

Tabel 3. Video Uji Coba Aplikasi

Video AI	Judul Video
	miaw miaw miaw song sad (lyrics video visual)

Video AI	Judul Video
	AI interviews the people onboard the Titanic
	Google's new AI video tool Veo 3 is WILD!
	We interviewed SKYRIM characters! (PART 2)
	Will Smith Eating a spaghetti in 2025 4K

2.3 Arsitektur Model

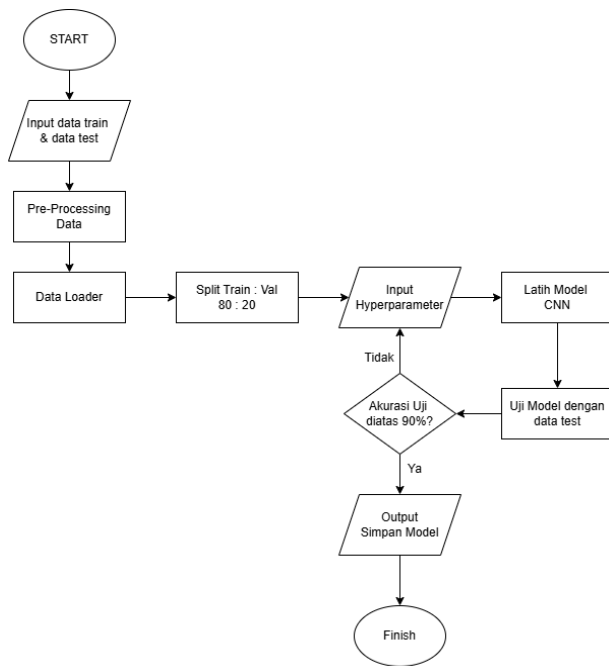
Model penelitian ini menggunakan arsitektur *ResNet-50* yang diimplementasikan dengan PyTorch. Arsitektur ini digunakan karena dapat menghasilkan akurasi yang tinggi. Berikut merupakan alur arsitekturnya:

- Input Layer:** Menerima gambar *dataset* yang sudah diubah ukuran menjadi 224×224 , sudah diaugmentasi, dan sudah dinormalisasi sesuai dengan *ImageNet* normalisasi.
- Backbone (ResNet-50):** Merupakan bagian utama, menggunakan model *pretrained* dari *ImageNet* untuk menerapkan *transfer learning*. Bagian ini akan secara otomatis mengekstrak fitur-fitur visual dari gambar melalui *blok* konvolusi.
- Classifier Head:** Fitur gambar yang sudah diekstrak akan diproses oleh *Global Average Pooling* untuk mereduksi dimensi, diikuti oleh sebuah *Fully Connected Layer*.
- Output Layer:** Sebuah lapisan *Dense* terakhir dengan aktivasi *Softmax* menghasilkan *output* berupa probabilitas untuk 2 kelas, yaitu gambar AI dan gambar asli.

2.4 Alur Pelatihan Model

Alur proses pelatihan model deteksi gambar AI dan gambar asli menggunakan algoritma CNN. Berikut Gambar 9 merupakan *flowchart* serta penjelasan setiap langkah dalam *flowchart*:

- Start**
Proses dimulai dengan pengumpulan data.
- Input Data Train & Data Test**
Memasukkan *dataset* kedalam program.
- Pre-processing Data**
Data citra yang telah dikumpulkan akan diproses, seperti *resize* gambar, konversi citra ke Tensor, normalisasi sesuai *ImageNet*, dan augmentasi *flip horizontal*.
- Data Loader**
Setelah dilakukan *preprocessing*, data akan di masukkan ke dalam model secara *batch* yang dapat diatur.
- Pembagian Data Validasi**
Data validasi itu diambil dari *data train*. Jadi *data train* nanti dibagi 20% untuk Data Validasi.
- Menentukan Hyperparameter**
Hyperparameter CNN (misalnya *learning rate*, *optimizer*, *epoch*, *dll*) ditentukan untuk mendapatkan hasil klasifikasi terbaik.
- Melatih Model CNN**
Model CNN dilatih menggunakan *data train*.
- Uji Model**
Model di uji menggunakan *data Test* dan dihitung akurasi uji.
 - Apakah Akurasi Uji di atas 90%?
 - Jika akurasi model lebih dari 90%, model dianggap optimal.
 - Jika akurasi kurang dari 90%, maka dilakukan penyesuaian *hyperparameter* dan model dilatih ulang.
- Mengubah Hyperparameter**
Jika akurasi belum mencapai target, dilakukan pengubahan *hyperparameter* dan model kembali dilatih.
- Simpan Model Optimal**
Model yang telah mencapai akurasi optimal akan disimpan untuk digunakan dalam sistem aplikasi pengenalan gambar AI atau bukan.
- Finish**
Proses selesai, model siap digunakan untuk sistem aplikasi pengenalan gambar AI atau bukan.



Gambar 9. Flowchart Pelatihan Model

2.5 Penilaian Performa Model

Penilaian performa model CNN dilakukan dengan cara evaluasi menggunakan berbagai metrik untuk mengukur efektivitas model. Metrik yang digunakan adalah *Confusion Matrix*. *Confusion matrix* adalah suatu metode yang digunakan untuk mengukur parameter klasifikasi dengan mencari akurasi, *loss* dan presisi[9]. Berikut 4 istilah setiap parameter seperti pada Tabel 4 di bawah ini.

Tabel 4. Confusion Matrix

Prediksi		
Aktual	True	False
True	TP	FP
False	FN	TN

Keterangan :

- True Positif* (TP) : Data positif yang diprediksi dengan benar.
- True Negatif* (TN) : Data negatif yang diprediksi dengan benar.
- False Positif* (FP) : Data negatif yang diprediksi sebagai data positif.
- False Negatif* (FN) : Data positif yang diprediksi sebagai data negatif.

Dari *confusion matrix*, bisa mendapatkan metrik performa, seperti akurasi, *precision*, dan *recall*, yang penting untuk proses evaluasi efektivitas model klasifikasi gambar AI:

- Akurasi : Nilai hasil yang benar pada sebuah sistem untuk mengenali *input data* agar *output* yang

dihasilkan oleh sistem dapat dievaluasi. Berikut persamaan Akurasi.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

- Precision* : Tingkat ketepatan sistem dalam memberikan informasi agar sesuai dengan informasi yang diminta. Berikut persamaan *Precision*.

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

- Recall* : Tingkat pengukuran sistem menemukan info yang sesuai. Berikut persamaan *recall*.

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

3. HASIL DAN PEMBAHASAN

3.1 Hasil Eksperimen

Pada bagian ini akan dilakukan eksperimen terhadap beberapa parameter, untuk menguji parameter mana yang nantinya memiliki akurasi uji tertinggi. Model dengan akurasi tertinggi akan disimpan lalu dijadikan model utama untuk GUI. Parameter diubah secara manual oleh penulis dan parameter yang diubah itu di antara lain *optimizer*, *epoch*, dan *learning rate*. Untuk lebih jelasnya bisa dilihat pada Tabel 5 Evaluasi Eksperimen Model.

Tabel 5. Evaluasi Eksperimen Model

Optimizer	Epoch	Learning rate	Akurasi Uji
Adam	5	0,001	79,88%
Adam	15	0,001	88,89%
AdamW	10	0,002	76,98%
SGD	8	0,002	90,29%
SGD	15	0,002	97,30%

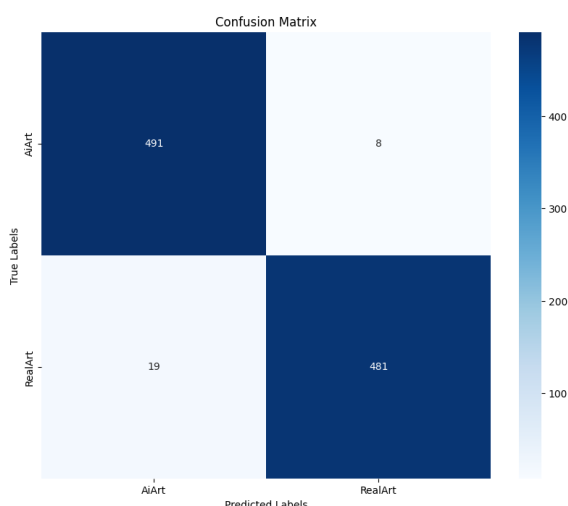
Dari tabel tersebut didapatkan hasil akurasi tertinggi adalah dengan menggunakan parameter *optimizer* SGD, *epoch* 15, dan *learning rate* 0,002 dengan akurasi mencapai 97,30%. Karena memiliki akurasi tertinggi maka model inilah yang akan disimpan lalu digunakan pada aplikasi. Pada Tabel 6 menunjukkan detail hasil eksperimen pada *epoch* terakhir dari setiap eksperimen, mulai dari akurasi latih, akurasi validasi, hasil *precision*, dan hasil *recall* dengan data uji. Urutannya sama dengan urutan pada Tabel 5.

Tabel 6. Detail Hasil Eksperimen

Eksperimen	Akurasi		Precision		Recall	
	Latih	Validasi	AI	Asli	AI	Asli
Adam	87,90%	87,72%	0,75	0,88	0,91	0,69
Adam	93,64%	86,63%	0,84	0,96	0,96	0,81
AdamW	86,76%	86,96%	0,70	0,91	0,94	0,60
SGD	95,19%	91,41%	0,85	0,97	0,98	0,83
SGD	98,59%	93,37%	0,96	0,98	0,98	0,96

Perbedaan tipis antara nilai akurasi latih dengan akurasi validasi pada model terbaik yang terpilih menunjukkan bahwa model telah belajar dengan baik dari *data train* dan mampu menggeneralisasi pengetahuannya pada data validasi yang belum pernah model lihat.

Untuk menganalisis kesalahan klasifikasi secara lebih visual, *confusion matrix* dibuat seperti yang ditunjukkan pada Gambar 10. Matriks tersebut memetakan perbandingan antara label prediksi dari model dengan label yang sesungguhnya pada model terbaik.



Gambar 10. Confusion Matrix Model Terbaik

Dari Gambar 10 dapat diamati bahwa, diagonal utama (491, 481) menunjukkan jumlah prediksi yang benar untuk setiap kelas. Angka-angka tersebut menunjukkan betapa baiknya performa model tersebut. Meskipun tetap ada gambar yang salah untuk diklasifikasi, terdapat 19 gambar asli yang diprediksi sebagai gambar AI dan terdapat 8 gambar AI yang diprediksi sebagai gambar asli.

3.2 GUI Deteksi Gambar AI

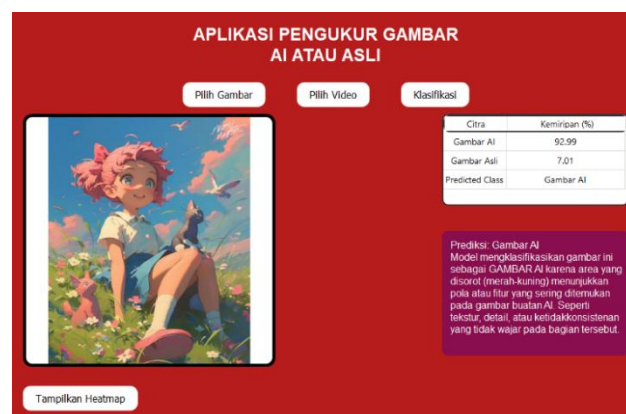
Berikut Gambar 11 ini adalah tampilan awal dari GUI klasifikasi gambar dan penjelasan setiap tombol pada aplikasi secara singkat.



Gambar 11. Tampilan Awal GUI

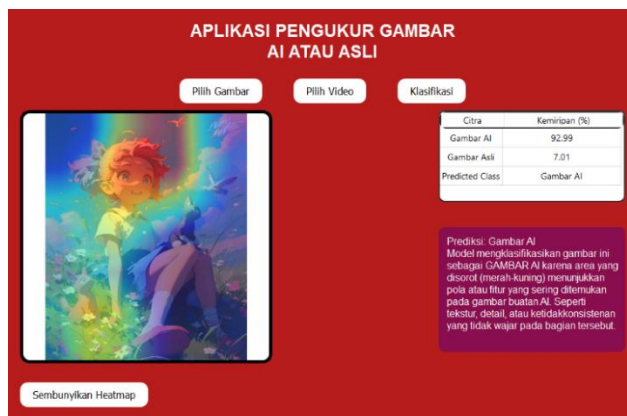
Di tampilan awal ini, pada bagian atas GUI terdapat tiga tombol utama dengan fungsi yang berbeda-beda. Tombol “Pilih Gambar” berfungsi untuk memilih gambar pada folder *device* yang mau diklasifikasi oleh aplikasi. Format gambar yang diunggah itu bisa ‘jpg’, ‘png’, ‘jpeg’ dan ‘bmp’. Tombol “Pilih Video” akan membuka *window* baru yang mencakup tombol yang diperlukan untuk mendeteksi video buatan AI.

Sedangkan tombol “Klasifikasi” seperti yang sudah tertulis pada tombolnya, tombol ini berfungsi untuk menjalankan proses klasifikasi pada gambar yang sudah diunggah. Hasil klasifikasinya akan muncul di bagian kanan gambar yang akan menampilkan akurasi prediksi pada gambar dalam bentuk tabel sekaligus memunculkan penjelasan singkat berdasarkan prediksi model. Hal itu bisa dilihat pada Gambar 12 Tampilan Hasil Klasifikasi.



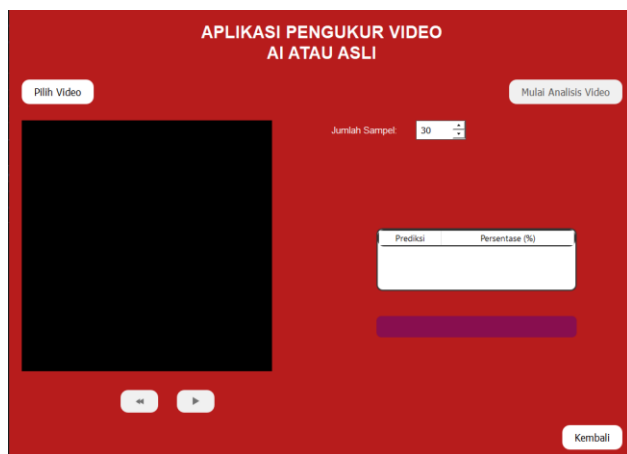
Gambar 12. Tampilan Hasil Klasifikasi

Di bawah *frame* tampilan gambar terdapat tombol “Tampilkan Heatmap” yang berfungsi untuk menunjukkan area mana yang paling mempengaruhi hasil klasifikasi sistem. Warna merah yang lebih intens pada *heatmap* menandakan area dengan kontribusi paling signifikan, yang menjadi dasar utama sistem dalam menentukan hasil klasifikasi. Untuk membuat hal tersebut saya menggunakan Grad-CAM (*Gradient-weighted Class Activation Mapping*) teknik yang memang sering digunakan untuk hal tersebut. Hasil *heatmap* bisa dilihat pada Gambar 13.

Gambar 13. Tampilan *Heatmap* Gambar Klasifikasi

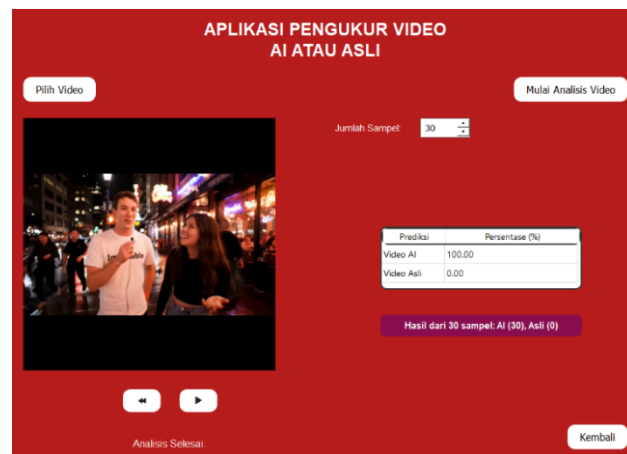
3.3 GUI Deteksi Video AI

Berikut Gambar 14 ini adalah tampilan *window* dari GUI deteksi video AI dan penjelasan setiap tombol pada aplikasi secara singkat.



Gambar 14. Tampilan GUI Klasifikasi Video

Pada tampilan di atas GUI terdapat 2 tombol utama dengan fungsi yang berbeda-beda. Tombol “Pilih Video” berfungsi untuk memilih video pada folder *device* yang mau diklasifikasi oleh aplikasi. Format video yang diunggah itu bisa berupa ‘mp4’, ‘avi’, ‘mov’, dan ‘mkv’. Setelah video diunggah, video dapat diputar dengan menekan tombol ‘play’ yang ada di bawah *frame* video. Mekanisme deteksi video pada aplikasi ini adalah video yang diunggah akan diambil *sample-sample* gambarnya. Jadi masih bisa dibilang deteksi gambar juga. Sebelum memulai proses analisis, pengguna dapat memilih seberapa banyak *sample* gambar yang ingin diambil. Setelah menentukan hal tersebut, pengguna tinggal menekan tombol “Mulai Analisis Video” untuk memulai proses analisis.



Gambar 15. Tampilan Hasil Klasifikasi Video

Bisa dilihat pada Gambar 15, *sample* gambar akan dideteksi lalu akan dihitung probabilitasnya. Hasil probabilitas akan dimunculkan pada tabel di sebelah kanan dan di bawahnya terdapat jumlah gambar *sample* yang diklasifikasi AI dan berapa *sample* gambar yang dikira asli. Untuk kembali ke *window* GUI deteksi gambar, pengguna bisa menekan tombol “Kembali” di pojok kanan bawah.

3.4 Hasil Uji Coba Aplikasi

Pertama, sistem aplikasi yang sudah dibuat akan diuji dengan gambar yang sudah dikumpulkan melalui *website* Pinterest, seperti yang ditunjukkan pada Tabel 2 memasukkan semua gambar tersebut satu-satu ke dalam sistem aplikasi, untuk melihat hasil perhitungan model yang digunakan pada aplikasi. Untuk urutan gambar yang dimasukkan itu sama dengan urutan pada Tabel 2 hasil perhitungannya bisa dilihat pada Tabel 7.

Tabel 7. Evaluasi Aplikasi Deteksi Gambar

Kelas Asli	Kemiripan (%)		Prediksi
	AI	Asli	
AI	92,99	7,01	AI
AI	99,70	0,30	AI
AI	62,68	37,32	AI
AI	92,43	7,57	AI
AI	99,70	0,30	AI
Asli	15,65	84,35	Asli
Asli	2,87	97,13	Asli
Asli	67,84	32,16	AI
Asli	10,14	89,84	Asli
Asli	34,32	65,68	Asli

Berdasarkan Tabel 6 performa sistem aplikasi pengukur gambar AI atau asli sudah sangat baik dalam mengenali gambar yang bukan dari *dataset*. Akurasi yang didapat itu

mencapai 90%. Memang masih ada gambar yang salah dikenali, tetapi nilai perbandingannya itu tidak terlalu mutlak ke satu sisi. Karena hal itu, menurut penulis sistem tetap layak disebut “BERHASIL” dalam mengenali gambar AI atau gambar asli.

Kedua, sistem aplikasi juga akan diuji dengan video yang sudah dikumpulkan melalui platform YouTube. Seperti yang ditunjukkan pada Tabel 3 memasukkan semua video tersebut satu-satu ke dalam sistem aplikasi, untuk melihat hasil perhitungan model yang digunakan pada aplikasi. Untuk urutan video yang dimasukkan itu sama dengan urutan pada Tabel 3 hasil perhitungannya bisa dilihat pada Tabel 8.

Tabel 8. Evaluasi Aplikasi Deteksi Video

Judul Video AI	Jumlah Sampel	Persentase (%)		Prediksi
		AI	Asli	
<i>miaw miaw miaw song sad (lyrics video visual)</i>	30	100	0	AI
<i>AI interviews the people onboard the Titanic</i>	30	93,33	6,67	AI
<i>Google's new AI video tool Veo 3 is WILD!</i>	25	100	0	AI
<i>We interviewed SKYRIM characters! (PART 2)</i>	30	60	40	AI
<i>Will Smith Eating a spaghetti in 2025 4K</i>	25	88	12	AI

Berdasarkan Tabel 8 performa sistem aplikasi identifikasi video AI sudah sangat baik dalam mengenali gambar-gambar *sample* yang didapat dari video. Akurasi yang didapat itu mencapai 100%. Dengan hasil nilai seperti itu, menurut penulis sistem layak disebut “BERHASIL” dalam mengenali video buatan AI.

4. KESIMPULAN

Penelitian ini berhasil membuat suatu sistem yang dapat mengenali media buatan AI menggunakan algoritma CNN dan membuat sistem aplikasi yang interaktif untuk mendeteksi media buatan AI. Berdasarkan hasil eksperimen, model yang dikembangkan menunjukkan tingkat akurasi tertinggi sebesar 97,30% menggunakan parameter *optimizer* SGD, *epoch* 15, dan *learning rate* 0.002. Untuk hasil performa akurasi sistem aplikasi untuk mengukur gambar AI atau asli mencapai 90%. Sedangkan hasil performa akurasi aplikasi untuk mengidentifikasi video AI mencapai 100%.

Aplikasi berbasis GUI juga telah dikembangkan menggunakan PyQt5 di Python. Aplikasi ini memungkinkan pengguna untuk mengunggah media buatan AI dan mendapatkan hasil klasifikasi beserta persentase kemiripan media tersebut dengan media buatan AI atau

media asli. Sistem ini dapat menjadi referensi dalam pengembangan sistem AI yang lebih akurat dan efisien.

Meskipun performanya cukup baik masih ada ruang untuk dikembangkan. Seperti menggunakan sepenuhnya *dataset* yang dikumpulkan secara manual bukan dari *dataset* Kaggle yang merupakan *dataset skunder*. Untuk pengawasan dan penyesuaian gambar yang lebih baik.

DAFTAR PUSTAKA

- [1] J. Qin, W. Pan, X. Xiang, Y. Tan, and G. Hou, “A biological image classification method based on improved CNN,” *Ecol. Inform.*, vol. 58, p. 101093, Jul. 2020, doi: 10.1016/j.ecoinf.2020.101093.
- [2] R. Xin, J. Zhang, and Y. Shao, “Complex network classification with convolutional neural network,” *Tsinghua Sci. Technol.*, vol. 25, no. 4, pp. 447–457, Aug. 2020, doi: 10.26599/TST.2019.9010055.
- [3] S. Adventino Gulo, A. Amelia Pertiwi, S. Putri Syaifullah Nasution, and H. Syahputra, “DETEKSI DEEPPAKE DALAM CITRA MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN),” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 9, no. 5, pp. 8655–8660, Jul. 2025, doi: 10.36040/jati.v9i5.14896.
- [4] M. H. V. Sinaga, M. Albirra, and M. F. Sidiq, “Klasifikasi Gambar Pemandangan dengan Kecerdasan Buatan Berbasis CNN,” *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 2, pp. 412–417, Apr. 2024, doi: 10.35870/jtik.v8i2.1424.
- [5] A. Prayoga, Maimunah, P. Sukmasetya, Muhammad Resa Arif Yudianto, and Rofi Abul Hasani, “Arsitektur Convolutional Neural Network untuk Model Klasifikasi Citra Batik Yogyakarta,” *J. Appl. Comput. Sci. Technol.*, vol. 4, no. 2, pp. 82–89, Nov. 2023, doi: 10.52158/jacost.v4i2.486.
- [6] K. Azmi, S. Defit, and S. Sumijan, “Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat,” *J. UNITEK*, vol. 16, no. 1, pp. 28–40, Jun. 2023, doi: 10.52072/unitek.v16i1.504.
- [7] M. A. N. Hidayat, Kusrini, and Hanafi, “Convolutional Neural Network Pada Identifikasi Varian Tanaman Anggur Menggunakan Resnet-50,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 10, no. 3, pp. 61–70, 2023, doi: https://doi.org/10.35957/jatisi.v10i3.4789.
- [8] T. Rhyosvaldo, Aurellio and P. Ricky, Eka, “Penerapan Metode Deep Learning Menggunakan Algoritma CNN Dengan Arsitektur VGG Net Untuk Pengenalan Cuaca,” *J. Informatics Comput. Sci.*, vol. 05, no. 01, pp. 48–57, Jul. 2023, doi: 10.26740/JINACS.V5N01.P48-57.
- [9] Dian Anisa Agustina, “Klasifikasi Citra Jenis Kulit Wajah Dengan Algoritma Convolutional Neural Network (Cnn) Resnet-50,” *J. Ris. Sist. Inf.*, vol. 1,

- no. 3, pp. 01–07, Jul. 2024, doi: 10.69714/13sbby24.
- [10] U. Ungkawa and G. AL Hakim, “Klasifikasi Warna pada Kematangan Buah Kopi Kuning menggunakan Metode CNN Inception V3,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 11, no. 3, p. 731, Jul. 2023, doi: 10.26760/elkomika.v11i3.731.
- [11] M. M. Taye, “Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions,” *Computation*, vol. 11, no. 3, p. 52, Mar. 2023, doi: 10.3390/computation11030052.
- [12] M. R. Efrian and U. Latifa, “IMAGE RECOGNITION BERBASIS CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK MENDETEKSI PENYAKIT KULIT PADA MANUSIA,” *Power Elektron. J. Orang Elektro*, vol. 11, no. 2, p. 276, Jul. 2022, doi: 10.30591/polektro.v12i1.3874.
- [13] T. Berliani, E. Rahardja, and L. Septiana, “Perbandingan Kemampuan Klasifikasi Citra X-ray Paru-paru menggunakan Transfer Learning ResNet-50 dan VGG-16,” *J. Med. Heal.*, vol. 5, no. 2, pp. 123–135, Aug. 2023, doi: 10.28932/jmh.v5i2.6116.
- [14] A. Raup, W. Ridwan, Y. Khoeriyah, Q. Yulianti Zaqiah, and U. Islam Negeri Sunan Gunung Djati Bandung, “Deep Learning dan Penerapannya dalam Pembelajaran,” *JIIIP - J. Ilm. Ilmu Pendidik.*, vol. 5, no. 9, pp. 3258–3267, Sep. 2022, doi: 10.54371/JIIP.V5I9.805.
- [15] F. Rizki, M. P. Kharisma Putra, M. A. Assuja, and F. Ariany, “Implementasi Deep Lering Lenet Dengan Augmentasi Data Pada Identifikasi Anggrek,” *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 4, no. 3, pp. 357–366, Sep. 2023, doi: 10.33365/jatika.v4i3.3652.
- [16] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, “Transfer learning: a friendly introduction,” *J. Big Data*, vol. 9, no. 1, p. 102, Oct. 2022, doi: 10.1186/s40537-022-00652-w.