



InComTech: Jurnal Telekomunikasi dan Komputer
vol.15, no.3, Desember 2025, 228-242
<http://publikasi.mercubuana.ac.id/index.php/Incomtech>
P-ISSN: 2085-4811 E-ISSN: 2579-6089

Implementasi *Convolutional Neural Network* dengan *Transfer Learning* Inception-V3 untuk Membatasi Web Pornografi

Mukijo*, Syahid Abdullah, Riad Sahara

*Program Studi PJJ Informatika, Universitas Siber Asia,
Jl. Harsono RM, Ragunan, Pasar Minggu Jakarta Selatan*

Email Penulis Koresponden:

*mukijo@student.unsia.ac.id

Abstrak :

Penyebaran pornografi melalui internet di masyarakat masih marak hingga saat ini dan dapat menimbulkan dampak negatif bagi penggunaannya. Web pornografi semakin hari semakin berkembang dan bervariasi, demikian juga metode untuk mengaksesnya. Sistem yang efektif untuk membatasi peredaran pornografi menjadi kebutuhan penting untuk melengkapi sistem yang sudah ada. *Convolutional Neural Network* (CNN) merupakan sebuah algoritma klasifikasi yang dapat dimanfaatkan untuk mendeteksi web pornografi berdasarkan gambar yang ada pada halaman web tersebut. Algoritma CNN dipilih karena kemampuannya yang menjanjikan dalam mengekstraksi fitur gambar. Selain itu dengan mengimplementasikan *transfer learning*, proses pembuatan model bisa dilakukan dengan cepat, efektif dan efisien. Penelitian ini bertujuan untuk membangun model klasifikasi gambar dan mengimplementasikan model ke dalam aplikasi berbasis web di sisi pengguna. Metode yang digunakan dalam penelitian ini meliputi klasifikasi citra, *transfer learning*, *web scrapping* dan *prototyping* perangkat lunak. Eksperimen berhasil membangun model klasifikasi citra pornografi-non pornografi dengan akurasi 99.15%. Model kemudian diimplementasikan ke dalam aplikasi ekstensi browser Chrome agar dapat dilakukan klasifikasi secara otomatis. Pengujian terhadap aplikasi yang dibangun menunjukkan aplikasi bekerja cukup baik dan mampu memblokir sebagian besar halaman web dengan konten pornografi.

This is an open access article under the [CC BY-NC](#) license



Kata Kunci:

*Deep Learning;
Convolutional Neural Network;
Transfer Learning;
Klasifikasi Gambar;
Pornografi.*

Riwayat Artikel:

Diserahkan 24 Juli 2024

Direvisi 22 Agustus 2025

Diterima 26 Agustus 2025

DOI:

10.22441/incomtech.v15i3.30299

1. PENDAHULUAN

Aksesibilitas internet tidak bisa dipungkiri memunculkan berbagai masalah, termasuk penyebaran konten pornografi yang tidak terkontrol. Situs-situs web yang berisi konten pornografi dapat dengan mudah diakses oleh siapa pun, termasuk mereka yang masih di bawah umur. Pornografi memiliki dampak merusak bagi pengaksesnya [1]. Kalangan remaja dan anak muda merupakan pihak yang paling rentan terhadap paparan pornografi [2]. Dalam dunia pekerjaan, pornografi dapat mempengaruhi produktivitas kerja dan perusahaan [3]. Penyebaran konten pornografi secara umum menyebabkan pelanggaran privasi dan pencemaran lingkungan digital.

Usaha untuk membatasi penyebaran pornografi di internet sudah banyak dilakukan baik oleh pemerintah maupun penyedia layanan internet. Sebagai contoh sepanjang tahun 2018 Kementerian Komunikasi dan Informatika (Kominfo) telah menangani 984.441 konten negatif yang tersebar di media online. Dari jumlah tersebut, terdapat 91% di antaranya adalah konten pornografi [4]. Metode yang saat ini paling banyak dipakai dalam mencegah akses kepada konten pornografi di internet dapat dibagi ke dalam 4 kelompok : (1) berdasarkan kata di dalam alamat url, (2) database alamat url, (3) berdasarkan kata-kata di dalam konten website, dan (4) *DNS Filtering* [1, 5, 6]. Metode-metode tradisional tersebut memiliki banyak kekurangan dan ketidakefektifan karena penyedia konten web terus berkembang dan bervariasi [7]. Metode lain yang sampai saat ini masih berkembang di antaranya adalah deteksi pornografi berdasarkan citra dengan berbasis pada kulit [8-10] dan berdasarkan bentuk seperti organ sensitif [11].

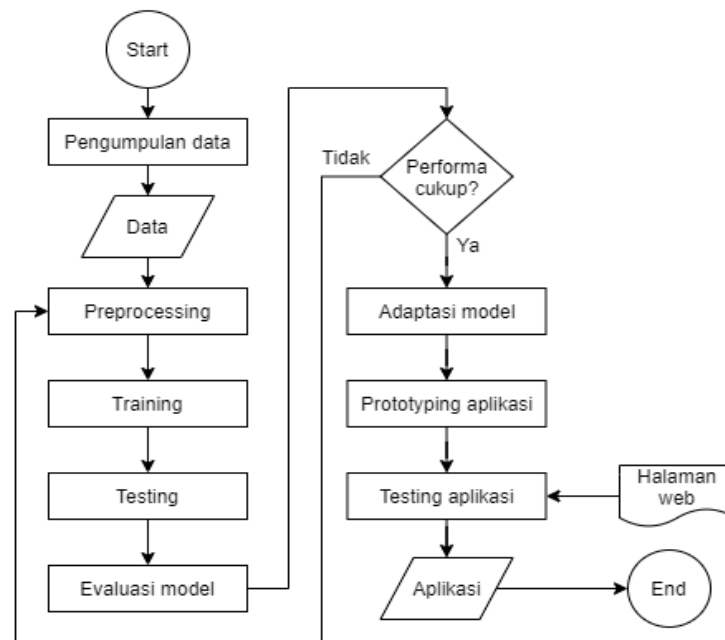
Convolutional Neural Network (CNN) merupakan algoritma klasifikasi gambar yang dapat dimanfaatkan untuk mendeteksi citra pornografi. Beberapa penelitian yang sudah dilakukan sebelumnya menunjukkan kemampuan CNN dalam melakukan klasifikasi citra pornografi dengan hasil yang cukup menjanjikan dengan tingkat akurasi 78-94.24 % [12, 13] dan masih terbuka banyak peluang untuk terus dikembangkan lebih lanjut. Pada penelitian ini, penulis mengusulkan implementasi teknologi CNN untuk mengembangkan model klasifikasi gambar dan mengimplementasikannya ke dalam sistem otomatis untuk mengenali dan membatasi konten pornografi secara langsung.

Pembatasan dari sisi pengguna diharapkan mampu memberikan solusi alternatif untuk meminimalkan akses terhadap konten pornografi. Model klasifikasi dengan tingkat akurasi yang tinggi diharapkan mampu membedakan apakah sebuah gambar merupakan gambar pornografi atau bukan. Setelah didapatkan tingkat akurasi yang terpercaya, model diadaptasi agar dapat dijalankan pada lingkungan web dengan memanfaatkan *library* TensorFlow.js. Dengan metode *web scraping* yang berjalan pada aplikasi ekstensi Chrome, diharapkan elemen gambar pada halaman web dapat diambil dan dilakukan klasifikasi oleh model. Melalui pendekatan ini, diharapkan sistem mampu mengenali apakah halaman web merupakan web penyedia pornografi berdasarkan gambar yang ada pada halaman tersebut.

2. METODE

Penelitian ini berbentuk eksperimental. Metode eksperimental dalam ilmu komputer adalah metode yang diterapkan untuk mencari hubungan atau pengaruh antara variabel satu dengan variabel lain dalam kondisi yang terkendali dengan ketat [14]. Penelitian berbentuk eksperimental sangat diperlukan pada

pengembangan perangkat lunak. Sedangkan berdasarkan fungsinya, penelitian ini dikategorikan ke dalam penelitian dan pengembangan (*research and development*). Penelitian dan pengembangan adalah kegiatan pengumpulan, pengolahan, analisis, dan penyajian data yang dilakukan secara sistematis dan objektif yang disertai dengan kegiatan membangun sebuah produk untuk memecahkan suatu persoalan yang dihadapi [15]. Tahap penelitian dimulai dari pengumpulan data sampai dengan pengembangan perangkat lunak (gambar 1).



Gambar 1. Tahapan Penelitian

2.1 Pengumpulan Data

Data yang dipakai dalam penelitian ini termasuk ke dalam data sekunder, yaitu data yang telah ada dan tersedia sebelumnya. Dataset yang digunakan untuk pelatihan didapatkan dari internet meliputi beberapa sumber termasuk Kaggle, Figshare, dan hasil pencarian Google Image. Sumber data yang diperoleh dari internet sangat relevan dengan masalah yang ingin diteliti.

Sebanyak 15.437 gambar digunakan yang terdiri atas 2 kelas yaitu yang termasuk kategori pornografi dan non pornografi. Sebanyak 7.127 gambar adalah pornografi dan 8.310 lainnya adalah non pornografi. Jumlah tersebut diharapkan cukup representatif untuk mewakili gambar yang ada di internet dalam dua kelompok kategori.

2.2. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah jenis arsitektur *Deep Learning* yang secara khusus digunakan untuk tugas-tugas pengenalan dan klasifikasi gambar. CNN memiliki kemampuan untuk mengenali pola. Dalam cara kerjanya CNN meniru komunikasi sel-sel saraf dalam otak manusia, dengan neuron yang saling terhubung, dan memiliki arsitektur yang serupa. Yang membedakannya dari rangkaian saraf lain adalah operasi konvolusional yang diterapkan pada setiap bagian dari input sebelumnya untuk mengekstraksi pola dan menghasilkan peta

fitur [16]. Algoritma CNN memiliki 3 komponen utama yaitu *Convolution Layer*, *Pooling Layer*, dan *Fully Connected Layer*.

1) *Convolution Layer*

Convolution Layer adalah proses di mana filter (*kernel*) diterapkan pada gambar masukan untuk menghasilkan peta fitur (*feature map*). Filter bergerak di seluruh gambar dan melakukan operasi matriks (*dot product*) pada bagian gambar yang sesuai dengan ukuran filter [17]. Secara umum formula pada proses konvolusi dituliskan sebagai berikut :

$$S(i,j) = (I * K)(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) \cdot K(m,n) \quad (1)$$

Di mana :

- I adalah matriks input.
- K adalah matriks kernel (filter)
- m dan n adalah tinggi dan lebar kernel
- M adalah jumlah baris pada kernel K.
- N adalah kolom pada kernel K
- i dan j adalah koordinat pada output feature map

2) *Pooling Layer*

Pooling Layer berfungsi untuk mengurangi dimensi spasial dari peta fitur yang dihasilkan oleh *Convolutional Layer*. Tujuan utama *Pooling Layer* adalah mengurangi kompleksitas data dan mempercepat proses komputasi dengan tetap mempertahankan informasi penting [18]. Secara umum formula untuk *average pooling* dituliskan sebagai berikut :

$$P(x,y) = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} F(x+i, y+j) \quad (2)$$

Di mana :

- F adalah peta fitur input.
- P adalah *pooling*
- m dan n adalah tinggi dan lebar *window pooling*
- x dan y adalah koordinat pada *output pooled map*
- i dan j adalah indeks pada *window pooling*

3) *Fully Connected Layer*

Fully Connected Layer adalah bagian akhir dari CNN yang bertanggung jawab untuk menghubungkan setiap neuron dari lapisan sebelumnya ke setiap neuron di lapisan selanjutnya. Layer ini mengambil hasil dari proses ekstraksi fitur oleh *Convolution Layer* dan *Pooling Layer* dan menyusunnya menjadi vektor fitur tunggal [19]. Kemudian, *Fully Connected Layer* menggunakan bobot dan bias untuk menghitung aktivasi neuron di lapisan terakhir, yang merupakan output dari model [20, 21]. Formula untuk *Fully Connected* secara umum dituliskan sebagai berikut :

$$z = Wx + b \quad (3)$$

Di mana :

- W adalah matriks bobot.
- x adalah input.

- b adalah bias.
- z adalah output sebelum diterapkan fungsi aktivasi.

2.3. Preprocessing

Preprocessing gambar adalah serangkaian teknik yang digunakan untuk mempersiapkan gambar sebelum diolah dengan algoritma CNN. Teknik *preprocessing* yang diterapkan meliputi transformasi, validasi, dan *resizing*. Pada tahap transformasi dilakukan konversi format gambar ke dalam format JPG. Tahap validasi meliputi pemeriksaan apakah gambar memiliki 3 channel (RGB) dan gambar tidak rusak. Pada tahap *resizing*, semua gambar diubah dimensinya menjadi 299x299 piksel untuk setiap channel. Hal ini dilakukan karena arsitektur model Inception V3 membutuhkan input dengan dimensi 299 piksel \times 299 piksel \times 3 (RGB) yang merupakan hasil dari kompromi antara akurasi dan efisiensi komputasi [22]. Semua teknik *preprocessing* menggunakan *library* OpenCV pada bahasa pemrograman Python.

2.4. Training

Transfer Learning adalah metode yang menggunakan model yang sudah dilatih sebelumnya dan menggunakannya sebagai titik awal untuk mempelajari tugas baru [23]. Model yang sebelumnya dilatih dengan data raksasa dapat dilatih kembali untuk tugas baru dengan jumlah data yang lebih kecil namun tetap mendapatkan keakuratan dari pelatihan sebelumnya.

Menerapkan *transfer learning* memberikan banyak keuntungan dari segi waktu maupun sumber daya. Arsitektur jaringan tidak harus dirancang dari awal karena arsitektur yang ada sudah terbukti efektif. Data latih yang digunakan bisa ditekan lebih kecil karena model sudah mempelajari banyak fitur. Bagian *classifier* bisa disesuaikan menurut kebutuhan

Struktur model terlatih Inception-V3 [24] digunakan sebagai *base model* dengan lapisan *Fully Connected* dihilangkan. Inception-V3 merupakan struktur model unggul dengan akurasi tinggi [25, 26], populer untuk *transfer learning* di berbagai aplikasi *deep learning*, berperforma tinggi, mampu menyeimbangkan akurasi tinggi dengan efisiensi komputasi, serta mampu mengatasi *overfitting* dan *vanishing gradients* [27].

Struktur ini kemudian dibekukan agar bobotnya tidak dapat dilatih dan dijadikan sebagai *base model*. *Base model* lalu dihubungkan dengan lapisan *Global Average Pooling 2D* untuk meratakan output dari *convolutional base*. Di belakangnya diambahkan lapisan *Dense* dengan 128 unit neuron dan aktivasi *ReLU*. Jumlah 128 neuron cukup besar untuk menjembatani output dari Inception v3 sebanyak 2048 fitur ke dalam 2 kelas. Jumlah yang terlalu besar akan memperlambat proses *training* [28]. Fungsi aktivasi *ReLU* memiliki sparsity yang baik, kecepatan konvergensi yang cepat, perhitungan yang sederhana, dan secara efektif memecahkan masalah dispersi gradien [29]. *Dropout* 0.5 ditambahkan untuk mencegah *overfitting* dan meningkatkan generalisasi dengan menghilangkan sebagian neuron secara acak pada setiap *epoch*. Akibatnya model belajar dari banyak sub jaringan berbeda dan model tidak bergantung sepenuhnya pada neuron tertentu, mendorong pembelajaran yang lebih mandiri. Nilai 0.5 merupakan nilai yang optimal dalam berbagai jaringan dan tugas [30]. Pada lapisan terakhir, ditambahkan lapisan *Fully Connected* dan aktivasi *sigmoid* untuk output klasifikasi

biner (*porn* dan *not porn*). Fungsi aktivasi *sigmoid* dipakai pada layer output karena model melakukan klasifikasi biner di mana fungsi *sigmoid* mengubah nilai dalam rentang probabilitas 0 hingga 1 [31]. Struktur model dikompilasi dengan *optimizer Adam* dan *learning rate* 0.001. Optimizer Adam menggabungkan keunggulan dari algoritma optimisasi AdaGrad dan RMSProp, yang secara dinamis menyesuaikan laju peluruhan eksponensial untuk estimasi momen pertama dan estimasi momen kedua dalam memperbarui parameter. Algoritma Adam cocok digunakan untuk data dalam jumlah besar dan optimisasi fungsi objektif yang tidak stasioner dengan gradien yang bising (*noisy*) dan jarang (*sparse*) [32]. Adam bersifat *robust* dan sangat cocok untuk berbagai permasalahan optimisasi non-konveks dalam *machine learning* [33]. *Learning rate* mengatur bobot jaringan yang disesuaikan berdasarkan gradien dari fungsi *loss*, jika terlalu besar, nilai *training* akan berayun di sekitar nilai optimal sehingga rentan mengalami osilasi (menyimpang di kedua sisi titik ekstrem), sebaliknya jika terlalu kecil, kecepatan konvergensi menjadi lambat dan rentan mengalami *overfitting* [32]. Nilai 0.001 adalah nilai *default* Optimizer Adam yang baik dan banyak digunakan dalam berbagai kasus [33, 34]. Sebanyak 15.437 gambar dibagi menjadi 14.026 (90%) untuk data latih dan 1.411 (10%) untuk data uji. Model dilatih sebanyak 50 *epoch* dengan *batch size* 50. Nilai *batch size* rendah (16-64) banyak disarankan untuk mencapai performa model yang optimal [35, 36].

2.5. Testing dan Evaluasi

Pada evaluasi model digunakan fungsi *loss binary_crossentropy* untuk menghitung entropi silang antara label yang sebenarnya dengan prediksi yang dibuat oleh model, dan metrik *accuracy* untuk mengukur persentase prediksi yang benar dari total prediksi yang dibuat oleh model. Fungsi *loss binary_crossentropy* merupakan fungsi *loss* yang handal dan paling banyak digunakan dibandingkan fungsi *loss* lain dalam klasifikasi biner [37], dan merupakan fungsi *loss* yang khusus ditujukan untuk klasifikasi biner [38].

Untuk lebih memastikan akurasi dan mengurangi bias, diterapkan *10-fold cross validation*, yaitu data uji dibagi ke dalam 10 sub set kemudian akurasi dari setiap sub set dirata-ratakan. *10-fold cross validation* berguna untuk mengurangi bias agar model tidak condong ke kelas tertentu [39], menghindari *overfitting* jaringan saraf tiruan selama proses pelatihan [40], dan dapat meningkatkan akurasi [41]. Nilai 10 merupakan nilai mendekati optimal yang banyak dipakai dalam berbagai kasus *neural network* [39-41].

2.6. Adaptasi Model

Adaptasi model merupakan tahap konversi model dari format keras ke dalam format JSON. Untuk mengimplementasikan model di lingkungan web, model yang disimpan dalam format HDF5 perlu dikonversi ke format JSON menggunakan TensorFlow.js. Proses konversi ini memungkinkan model digunakan di lingkungan browser tanpa kebutuhan *server-side processing*.

2.7. Prototyping Aplikasi

Model *Prototyping* sebagai salah satu model SDLC (*Software Development Life Cycle*) adalah pendekatan populer dalam pengembangan perangkat lunak di mana prototipe dibangun, diuji, dan dikerjakan ulang hingga diperoleh prototipe yang

dapat diterima sebagai dasar untuk menghasilkan sistem atau perangkat lunak akhir [42]. Metode *prototyping* sesuai untuk pengembangan perangkat lunak secara cepat dan bertahap sehingga dapat langsung dievaluasi serta menghemat waktu dan sumber daya. Fungsi utama dari perangkat lunak diutamakan dan beberapa fungsi tambahan mungkin belum ditambahkan. *Prototyping* dalam penelitian ini secara umum dibagi ke dalam tiga tahap, yaitu perancangan, pembuatan program, dan evaluasi.

2.8. Web Scraping

Model klasifikasi membutuhkan gambar sebagai masukannya. Halaman web merupakan target dari penelitian ini. Agar model dapat diimplementasikan secara langsung pada halaman web, diperlukan teknik pengambilan gambar pada halaman web secara otomatis dengan teknik *web scraping* memanfaatkan DOM.

DOM (*Document Object Model*) adalah representasi data objek yang menyusun struktur dan konten dokumen halaman web. DOM mewakili dokumen sebagai node dan objek, sehingga bahasa pemrograman dapat berinteraksi dengan halaman tersebut. DOM disusun seperti struktur pohon yang terdiri atas *node* dan *childnodes* [43]. Dengan DOM, elemen gambar dapat diambil dengan memanggil *tag* ``.

2.9. Pengujian Aplikasi

Pengujian aplikasi dalam penelitian ini menggunakan pendekatan *Grey Box Testing*. *Grey Box Testing* merupakan perpaduan antara *Black Box Testing* dan *White Box Testing*. *Grey Box Testing* menguji perangkat lunak berdasarkan spesifikasi dengan menggunakan beberapa pengetahuan tentang kinerja internalnya [44]. Pengujian aplikasi berfokus pada fungsionalitas dan logika program untuk memastikan aplikasi secara keseluruhan berfungsi sesuai rencana.

3. HASIL DAN PEMBAHASAN

3.1. Performa Model

Pelatihan model dengan algoritma *Convolutional Neural Network* dengan *Transfer Learning* InceptionV3 berhasil meningkatkan akurasi model dibandingkan tanpa *Transfer Learning*. Pada pelatihan tanpa *Transfer Learning* didapatkan akurasi akhir sebesar 94,19%, sedangkan dengan *Transfer Learning* didapatkan akurasi akhir 99.15%. Akurasi sebesar 99.15% menunjukkan model berhasil menebak 99.15% dari keseluruhan data uji secara benar. Nilai akurasi yang tinggi dan *loss* yang rendah menandakan bahwa model berperforma sangat baik. Visualisasi nilai akurasi dan *loss* pada data latih maupun data uji ditunjukkan pada gambar 2 dan 3.

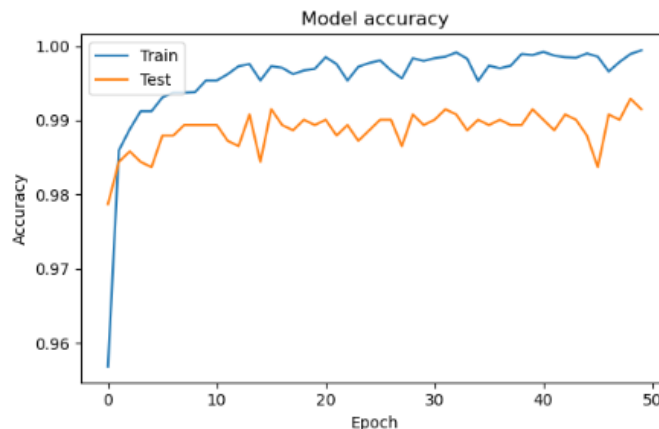
Formula akurasi dapat ditulis sebagai :

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Di mana :

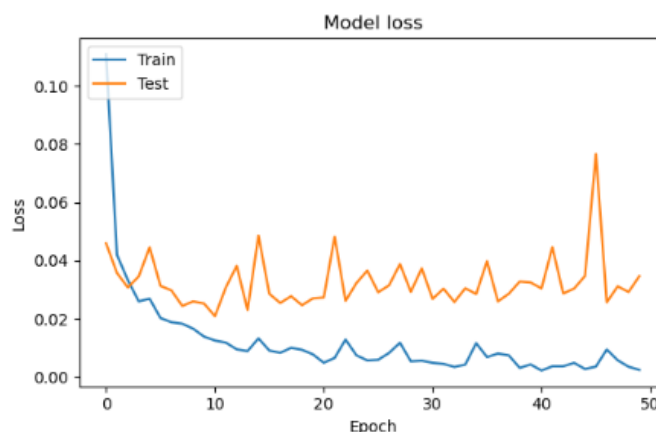
- TP: *True Positives* (jumlah data kelas *porn* yang diprediksi benar)
- TN: *True Negatives* (jumlah data kelas *not porn* yang diprediksi benar)
- FP: *False Positives* (jumlah data kelas *porn* yang diprediksi salah)

- FN: *False Negatives* (jumlah data kelas *not porn* yang diprediksi salah)



Gambar 2. Grafik Akurasi

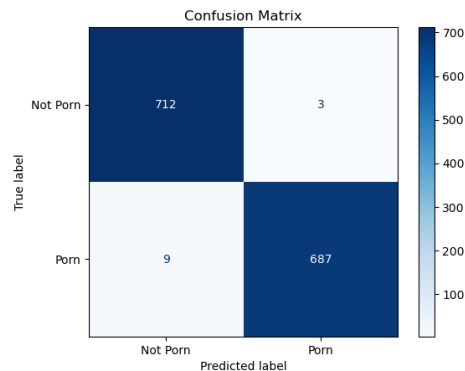
Sedangkan pada grafik *loss*, kurva menurun dan mencapai suatu titik stabil. Ini menunjukkan bahwa model menggeneralisasi dengan baik ke data yang tidak terlihat selama pelatihan. Kurva juga menunjukkan tidak terjadi *overfitting*. Grafik *loss* ditunjukkan pada gambar 3 di bawah.



Gambar 3. Grafik Loss

Validasi menggunakan *Confusion Matrix* memberikan gambaran jelas dan terperinci untuk mengevaluasi kinerja model klasifikasi, sejauh mana model mengklasifikasi dengan benar dan sejauh mana model melakukan kesalahan [45], memberi petunjuk kuat tentang hubungan antara kelas dan label yang mewakili makna semantik dan konsep yang ditetapkan untuk instans data [46], serta membantu memilih tindakan terbaik untuk meningkatkan kinerja model. *Confusion Matrix* juga membantu penulis untuk memutuskan *trade-off* sesuai kebutuhan aplikasi. *False Positif* harus ditekan sekecil mungkin karena akan berbahaya bagi aplikasi dan pengguna dengan menyebabkan alarm palsu (*false alarm*). Alarm palsu pada aplikasi akan menyebabkan halaman web yang sebenarnya aman menjadi terblokir dan pengguna menjadi terganggu. Di sisi lain *False Negatif* lebih bisa diterima dan tidak akan menyebabkan tindakan/gangguan apapun pada aplikasi dan pengguna.

Evaluasi dengan menggunakan *Confusion Matrix* pada penelitian ini menunjukkan sebagian besar data uji dapat diprediksi dengan benar untuk kelas *porn* maupun *not porn*. Visualisasi *Confusion Matrix* ditunjukkan pada gambar 4 di bawah.



Gambar 4. *Confusion Matrix*

Dari 1411 data uji, sebanyak 712 gambar bukan porno berhasil diprediksi secara benar ke dalam kelas *not porn* (*True Negative*) dan sebanyak 687 gambar porno berhasil diprediksi secara benar ke dalam kelas *porn* (*True Positive*). Sisanya sebanyak 9 gambar porno gagal diprediksi secara benar ke dalam kelas *porn* (*False Negative*) dan sebanyak 3 gambar bukan porno gagal diprediksi secara benar ke dalam kelas *not porn* (*False Positive*).

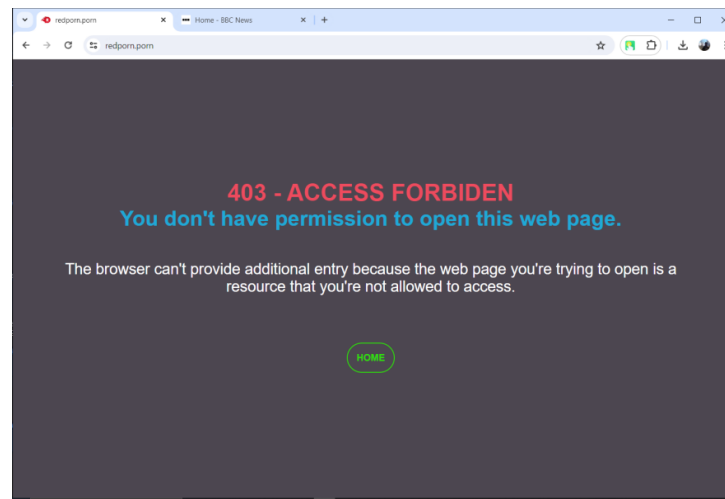
Secara umum model memiliki performa yang baik dalam mengklasifikasikan gambar porno dan bukan porno. Hasil pengujian model pada data baru menunjukkan bahwa model bekerja sangat baik dalam melakukan klasifikasi.

3.2. Performa Aplikasi

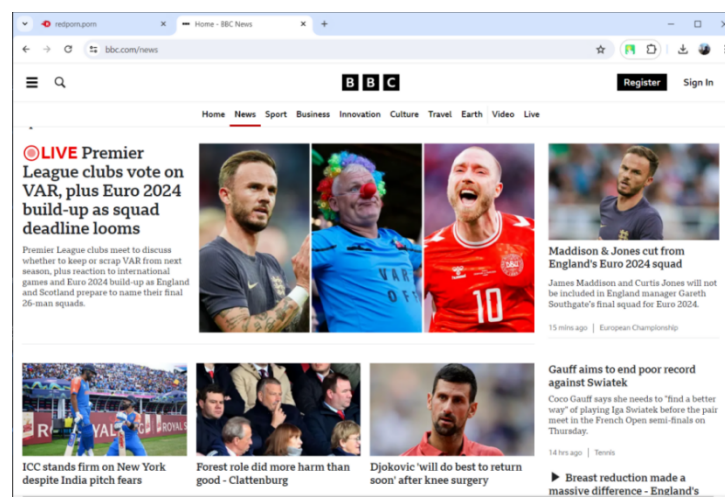
Aplikasi dirancang untuk mengambil gambar dari halaman web yang sedang dibuka, melakukan prediksi terhadap setiap gambar dengan model klasifikasi, dan melakukan pemblokiran terhadap halaman web. Sebuah nilai *threshold* ditetapkan untuk menentukan jumlah minimal gambar porno terdeteksi dalam sebuah halaman web untuk dilakukan pemblokiran. *Threshold* yang terlalu tinggi akan membuat halaman web pornografi yang memiliki sedikit gambar tidak terblokir, sebaliknya *threshold* yang terlalu rendah akan membuat web non pornografi lebih rawan terblokir akibat deteksi *False Positive*. Dengan asumsi rata-rata web pornografi memiliki cukup banyak gambar, dalam penelitian ini ditetapkan *threshold* 5 gambar terdeteksi. Angka 5 didasarkan pada serangkaian percobaan dengan nilai berbeda kemudian dianalisis untuk mendapatkan nilai dengan paling sedikit terjadi kesalahan.

Aplikasi yang dikembangkan memiliki kemampuan cukup baik dalam memblokir web pornografi dan tidak memblokir web non pornografi. Apabila web terdeteksi sebagai web pornografi, aplikasi akan melakukan pemblokiran dengan cara memanipulasi html. Tampilan halaman web terblokir ditunjukkan oleh gambar 5 di bawah. Sebaliknya, apabila web tidak terdeteksi sebagai web pornografi, aplikasi tidak akan melakukan tindakan lebih lanjut dan pengguna tetap bisa melanjutkan *browsing* tanpa gangguan apapun. Tampilan halaman web tidak

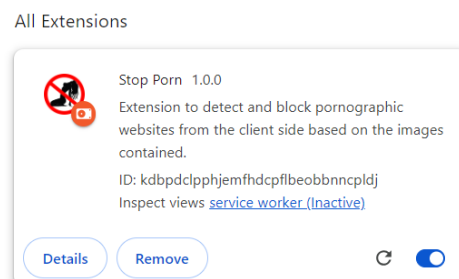
terblokir ditunjukkan oleh gambar 6 di bawah. Tampilan aplikasi ekstensi yang sudah terpasang pada peramban Google Chrome ditunjukkan oleh gambar 7.



Gambar 5. Tampilan halaman web terblokir



Gambar 6. Tampilan halaman web tidak terblokir



Gambar 7. Tampilan aplikasi terpasang

3.2. Pengujian Aplikasi

1) Pengujian *White Box* (integrasi)

Pengujian integrasi dilakukan untuk menguji logika program, yaitu bagaimana komponen-komponen dalam sistem saling bekerjasama membentuk satu kesatuan. Pengujian integrasi memastikan bahwa komponen-komponen tersebut dapat berkomunikasi dengan benar, mentransfer data dengan akurat, dan berfungsi secara harmonis. Pengujian dilakukan terhadap beberapa alur di dalam sistem yang diperlihatkan pada tabel 2 di bawah.

Tabel 1. Pengujian integrasi

Komponen	Logika Program	Hasil
content.js	Mengambil elemen gambar setelah halaman selesai dimuat	Sebagian besar berhasil
content.js	Mengirimkan url gambar ke background.js	Berhasil
background.js	Mengubah url ke ImageData	Sebagian besar berhasil
background.js	Memuat model	Berhasil
background.js	Melakukan prediksi	Berhasil
background.js	Mengirimkan respon ke content.js	Berhasil
content.js	Memanipulasi halaman web	Berhasil

Pada pengujian integrasi, terlihat bahwa sebagian kecil halaman web tidak berhasil diambil gambarnya. Pada sebagian kecil halaman web, url berhasil diambil namun bukan merupakan gambar yang diinginkan sehingga tidak berhasil diubah ke *ImageData*. Kegagalan pengambilan gambar menyebabkan halaman web tidak berhasil diblokir pada pengujian fungsional. Kegagalan pengambilan gambar dapat disebabkan oleh konfigurasi/logika halaman web yang tidak mengijinkan pengambilan gambar.

2) Pengujian *Black Box* (fungsionalitas)

Pengujian fungsionalitas dilakukan dengan memberikan input kepada aplikasi dan melihat apakah aplikasi memberikan output yang sesuai. Aplikasi menerima input berupa halaman web aktif, dan memberikan output halaman web yang termodifikasi secara html jika halaman web aktif diidentifikasi sebagai web pornografi dan tidak memberikan output apapun jika diidentifikasi sebagai web non pornografi. Pengujian yang dilakukan terhadap 100 alamat web dengan domain berbeda yang terdiri atas 50 halaman web pornografi (Kategori Barat, Jepang, Hentai, dan Indonesia) dan 50 halaman web non pornografi menunjukkan bahwa sebagian besar halaman web berhasil diidentifikasi dengan benar. Rangkuman daftar pengujian ditunjukkan pada tabel 1 di bawah.

Tabel 2. Pengujian fungsionalitas

Kategori web	Terblokir	Tidak terblokir
Pornografi	43	7
Non Pornografi	0	50

Semua web non pornografi berhasil diambil gambarnya dan tidak terblokir karena tidak memenuhi jumlah terdeteksi pornografi. Sedangkan pada halaman web pornografi, tidak semua terblokir dikarenakan gambar dari halaman tersebut tidak berhasil diambil. Aplikasi ekstensi browser Chrome (*Chrome extension*) ini telah dipublikasikan ke *Chrome Web Store* dengan nama Stop Porn.

4. KESIMPULAN

Proses pelatihan model dengan algoritma *Convolutional Neural Network* (CNN) dengan *Transfer Learning* Inception V3 terhadap 15.437 gambar berhasil membangun model klasifikasi dengan tingkat akurasi 99.15%. Penambahan *Transfer Learning* berhasil menaikkan tingkat akurasi hampir 5% dari 94.15 % dibandingkan tanpa menggunakan *Transfer Learning*. Berdasarkan evaluasi menggunakan *10-fold-cross-validation* dan *Confusion Matrix* disimpulkan model yang dihasilkan memiliki performa yang sangat baik.

Implementasi model klasifikasi ke dalam aplikasi dilakukan dengan mengembangkan aplikasi Ekstensi Chrome. Ekstensi bekerja mengambil gambar dari halaman web apapun yang sedang dibuka, melakukan *preprocessing* gambar, melakukan klasifikasi terhadap gambar, dan memanipulasi halaman web. Berdasarkan pengujian menggunakan metode *Grey Box*, didapatkan aplikasi bekerja dengan cukup baik. Uji coba terhadap 50 halaman web non pornografi yang di dalamnya terdapat cukup banyak gambar, aplikasi tidak melakukan pemblokiran satu halaman pun. Sedangkan uji coba terhadap 50 halaman web pornografi, aplikasi berhasil memblokir 43 halaman web. Sisa halaman web pornografi yang tidak berhasil diblokir dikarenakan gambar tidak berhasil diambil dapat dikarenakan konfigurasi halaman web bersangkutan.

REFERENSI

- [1] R. de Alarcón, J. I. de la Iglesia, N. M. Casado, and A. L. Montejo, "Online porn addiction: What we know and what we don't—a systematic review," *Journal of Clinical Medicine*, vol. 8, no. 1, p. 91, Jan. 2019. doi:10.3390/jcm8010091
- [2] H. Adarsh and S. Sahoo, "Pornography and its impact on adolescent/teenage sexuality," *Journal of Psychosexual Health*, vol. 5, no. 1, pp. 35–39, Jan. 2023. doi:10.1177/26318318231153984.
- [3] N. W. Mecham, M. F. Lewis-Western, and D. A. Wood, "The effects of pornography on unethical behavior in business," *Journal of Business Ethics*, vol. 168, no. 1, pp. 37–54, Jun. 2019. doi:10.1007/s10551-019-04230-8
- [4] V. Alhakim, S. Dewi, and A. Rompis, "Pembentukan Lembaga Independen Dalam pengawasan konten digital: Studi Komparasi Hukum antara Indonesia dengan Australia," *COMSERVA: Jurnal Penelitian dan Pengabdian Masyarakat*, vol. 3, no. 09, pp. 3627–3643, Jan. 2024. doi:10.59141/comserva.v3i09.1150
- [5] Y. Cheng, H. Jiang, Z. Zhang, Y. Du, and T. Chai, "Birds of a feather flock together: Generating pornographic and gambling domain names based on character composition similarity," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–17, Jul. 2022. doi:10.1155/2022/4408987.
- [6] Ulfah, N., Irawan, N. O., Nurfadila, P. D., Ristanti, P. Y., & Hammad, J. A. H. (2019). Blocking pornography sites on the internet private and University Access. *Bulletin of Social Informatics Theory and Application*, 3(1), 22–29. https://doi.org/10.31763/businta.v3i1.161
- [7] Wang, Z. et al. (2021) 'A new model for small target adult image recognition', *Procedia Computer Science*, 183, pp. 557–562. doi:10.1016/j.procs.2021.02.097.
- [8] M. S. Farooq et al., "Skin detection based pornography filtering using adaptive back

- propagation neural network,” 2019 *8th International Conference on Information and Communication Technologies (ICICT)*, Nov. 2019. doi:10.1109/iciict47744.2019.9001915.
- [9] H. A. Al Naffakh, R. Ghazali, N. K. El Abbadi, and A. N. Razzaq, “A review of human skin detection applications based on image processing,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 129–137, Feb. 2021. doi:10.11591/eei.v10i1.2497.
 - [10] D.-D. Phan et al., “A novel pornographic visual content classifier based on sensitive object detection,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, 2021. doi:10.14569/ijacsa.2021.0120591.
 - [11] H.-L. Tran et al., “Additional learning on object detection: A novel approach in pornography classification,” *Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications*, pp. 311–324, 2020. doi:10.1007/978-981-33-4370-2_22.
 - [12] Setiono, K, Kristian, Y, Gunawan. (2021) “Deteksi Citra Pornografi Memanfaatkan Deep Convolutional Neural Network”, *Insyst*, vol. 3(1), doi:10.52985/insyst.v3i1.172
 - [13] Putra, M.T.D, Ardiansyah, M.I, Aprianti, D. (2022) “Deteksi Konten Pornografi Menggunakan Convolutional Neural Network untuk Melindungi Anak dari Bahaya Pornografi”, *Jurnal Media Informatika Budidarma*, 6(4), pp. 2401-2409, DOI: <http://dx.doi.org/10.30865/mib.v6i4.4793>
 - [14] N. Rikatsih, dkk. “Metode Penelitian Bidang Ilmu Komputer,” in *Metode Penelitian di Berbagai Bidang*, Bandung, Jawa Barat: *Media Sains Indonesia*, 2021, pp. 3–5.
 - [15] S. Gustiani, “RESEARCH AND DEVELOPMENT (R&D) METHOD AS A MODEL DESIGN IN EDUCATIONAL RESEARCH AND ITS ALTERNATIVES,” *HOLISTICS JOURNAL*, vol. 1, no. 2, pp. 12–22, Dec. 2019.
 - [16] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of Convolutional Neural Networks: Analysis, applications, and prospects,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022. doi:10.1109/tnnls.2021.3084827.
 - [17] R. M. Prasmatio, B. Rahmat, and I. Yuniar, “DETEKSI DAN PENGENALAN IKAN MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK,” *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, vol. 1, no. 2, pp. 510–521, Jul. 2020.
 - [18] A. Zafar et al., “A comparison of pooling methods for Convolutional Neural Networks,” *Applied Sciences*, vol. 12, no. 17, p. 8643, Aug. 2022. doi:10.3390/app12178643.
 - [19] A. Dwi Putro and H. Tantyoko, “Hybrid algoritma VGG16-net dengan support vector machine untuk Klasifikasi Jenis Buah Dan Sayuran,” *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 5, no. 2, pp. 56–65, Jul. 2023. doi:10.35746/jtim.v5i2.335.
 - [20] D. Alamsyah and D. Pratama, “Implementasi convolutional neural networks (CNN) Untuk Klasifikasi Ekspresi Citra Wajah Pada Fer-2013 dataset,” *Jurnal Teknologi Informasi*, vol. 4, no. 2, pp. 350–355, Dec. 2020. doi:10.36294/jurti.v4i2.1714.
 - [21] L. Wang, C. Wang, Y. Li, and R. Wang, “Explaining the behavior of neuron activations in deep neural networks,” *Ad Hoc Networks*, vol. 111, p. 102346, Feb. 2021. doi:10.1016/j.adhoc.2020.102346.
 - [22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Jun. 2016. doi:10.1109/cvpr.2016.308
 - [23] S. Tammina, “Transfer learning using VGG-16 with deep convolutional neural network for classifying images,” *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, Oct. 2019. doi:10.29322/ijsrp.9.10.2019.p9420.
 - [24] Meena, G., Mohbey, K. K., & Kumar, S. (2023). Sentiment analysis on images using convolutional neural networks based inception-V3 transfer learning approach. *International Journal of Information Management Data Insights*, 3(1), 100174. <https://doi.org/10.1016/j.jjime.2023.100174>.
 - [25] M. Mujahid et al., “Pneumonia classification from X-ray images with inception-V3 and Convolutional Neural Network,” *Diagnostics*, vol. 12, no. 5, p. 1280, May 2022. doi:10.3390/diagnostics12051280).
 - [26] Y. F. Riti, R. P. Kristianto, and D. R. A. Setiawan, “PERFORM COMPARATION OF DEEP LEARNING METHODS IN GENDER CLASSIFICATION FROM FACIAL IMAGES”, *jitk*, vol. 10, no. 4, pp. 926–936, Jun. 2025
 - [27] C. Lin, L. Li, W. Luo, K. C. Wang, and J. Guo, “Transfer learning based traffic sign

- recognition using inception-V3 model,” *Periodica Polytechnica Transportation Engineering*, vol. 47, no. 3, pp. 242–250, Aug. 2018. doi:10.3311/pptr.11480
- [28] C. A. Doukim, J. A. Dargham, and A. Chekima, “Finding the number of hidden neurons for an MLP neural network using coarse to fine search technique,” *10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)*, May 2010. doi:10.1109/isspa.2010.5605430
- [29] Y. Bai, “Relu-function and derived function review,” *SHS Web of Conferences*, vol. 144, p. 02006, 2022. doi:10.1051/shsconf/202214402006
- [30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.
- [31] S. Sharma, S. Sharma, and A. Athaiya, “Activation functions in neural networks,” *International Journal of Engineering Applied Sciences and Technology*, vol. 04, no. 12, pp. 310–316, May 2020. doi:10.33564/ijeast.2020.v04i12.054
- [32] Y. Wang, Z. Xiao, and G. Cao, “A convolutional neural network method based on Adam optimizer with power-exponential learning rate for bearing fault diagnosis,” *Journal of Vibroengineering*, vol. 24, no. 4, pp. 666–678, Mar. 2022, doi: 10.21595/jve.2022.22271.
- [33] Kingma, D.P. and Ba, J. (2014) Adam: A Method for Stochastic Optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 14-16 April 2014.
- [34] L. Luo, Y. Xiong, Y. Liu, and X. Sun, “Adaptive gradient methods with dynamic bound of learning rate,” *arXiv preprint arXiv:1902.09843*, 2019.
- [35] I. Kandel and M. Castelli, “The effect of batch size on the generalizability of the Convolutional Neural Networks on a histopathology dataset,” *ICT Express*, vol. 6, no. 4, pp. 312–315, Dec. 2020. doi:10.1016/j.icte.2020.04.010
- [36] D. Masters and C. Luschi, “Revisiting small batch training for deep neural networks,” *arXiv preprint arXiv:1804.07612*, 2018.
- [37] K. Janocha and W. M. Czarnecki, “On loss functions for deep neural networks in classification,” *arXiv preprint arXiv:1702.05659*, 2017.
- [38] Usha Ruby Dr.A, Theerthagiri Prasannavenkatesan, J. J. Dr.I, and amsidhar Y. Dr., “Binary Cross entropy with deep learning technique for image classification,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 5393–5397, Aug. 2020. doi:10.30534/ijatcse/2020/175942020
- [39] K. Jung et al., “Evaluation of nitrate load estimations using neural networks and canonical correlation analysis with k-fold cross-validation,” *Sustainability*, vol. 12, no. 1, p. 400, Jan. 2020. doi:10.3390/su12010400.
- [40] G. Singh and R. K. Panda, “Daily sediment yield modeling with artificial neural network using 10-fold cross validation method: a small agricultural watershed, Kapgari, India,” *International Journal of Earth Sciences and Engineering*, vol. 4, no. 6, pp. 443–450, 2011.
- [41] S. M. Malakouti, “Improving the prediction of wind speed and power production of SCADA system with ensemble method and 10-fold cross-validation,” *Case Studies in Chemical and Environmental Engineering*, vol. 8, p. 100351, Dec. 2023. doi:10.1016/j.csee.2023.100351.
- [42] Senarath, U. S. (2021). Waterfall methodology, prototyping and agile development. *Tech. Rep.*, 1-16.
- [43] I. A. Ahmad Sabri, M. Man, W. A. Abu Bakar, and A. N. Mohd Rose, “Web data extraction approach for deep web using WEIDI,” *Procedia Computer Science*, vol. 163, pp. 417–426, 2019. doi:10.1016/j.procs.2019.12.124.
- [44] Z. Hamza and M. Hammad, “Testing approaches for web and mobile applications: An overview,” *International Journal of Computing and Digital Systems*, vol. 9, no. 4, pp. 657–664, Jul. 2020. doi:10.12785/ijcds/090413.
- [45] S. Sathyanarayanan and B. R. Tantri, “Confusion matrix-based performance evaluation metrics,” *African Journal of Biomedical Research*, pp. 4023–4031, Nov. 2024. doi:10.53555/ajbr.v27i4s.4345.
- [46] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, “Multi-label classifier performance evaluation with confusion matrix,” *Computer Science & Information Technology*, Jun. 2020. doi:10.5121/csit.2020.100801

