Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 8 ; Nomor 2 ; Juli 2025 ; Page 214-226
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

Deteksi Ketidakkonsistenan *Font* Sebagai Indikator Pemalsuan dan Penyuntingan Dokumen Digital Menggunakan *Convolutional Neural Network* (CNN)

Lumi Krismona[1], Annisa Ashari[2], Nurhayati[3], Adil Setiawan[4], Rika Rosnelly [5]

[1,2,3,4,5] Ilmu Komputer, Universitas Potensi Utama, Medan, Indonesia
Email: [1]lumiikrismona@gmail.com, [2]annisaashari19@gmail.com, [3]n.hayati2390@gmail.com, [4]adio165@gmail.com, [5]rika@potensi-utama.ac.id

Email Penulis Korespondensi: lumiikrismona@gmail.com

Article History:
Received Jun 15th, 2025
Revised Jun 30th, 2025
Accepted Jul 30th, 2025

Abstrak

Pemalsuan dan penyuntingan dokumen digital merupakan ancaman serius dalam konteks keamanan informasi dan validitas dokumen resmi. Salah satu contoh aktual adalah kasus pemalsuan dokumen kependudukan untuk Penerimaan Peserta Didik Baru (PPDB) 2024 oleh Ato et al.(Kompas, 2024), telah ditemukan penyalahgunaan data dan perubahan pada dokumen seperti Kartu Keluarga (KK) untuk memanipulasi zonasi pendidikan. Kasus ini menunjukkan bahwa dokumen digital sangat rentan dimanipulasi salah satunya melalui ketidakkonsistenan jenis *font* pada struktur dokumen digital. Penelitian ini bertujuan untuk mendeteksi secara otomatis terhadap ketidakkonsistenan font dalam dokumen digital menggunakan arsitektur *Convolutional Neural Network* (CNN). Model dilatih menggunakan 100.000 sampel dari *Document Font Recognition Dataset* (DTFR), dengan pra-pemrosesan berupa konversi *grayscale*, normalisasi dan *resize* citra menjadi 32×32 piksel. CNN dirancang dengan dua lapisan konvolusional, *max pooling*, *dropout* dan *dense layer*. Hasil evaluasi menunjukkan akurasi sebesar 96,85% dengan nilai *precision*, *recall* dan *F1-score* rata-rata sebesar 0,97. Pendekatan ini terbukti lebih akurat dibandingkan metode SVM yang sebelumnya dilaporkan hanya mencapai 94,6%. Penelitian ini menunjukkan bahwa pendekatan CNN efektif untuk mendeteksi ketidakkonsistenan *font* sebagai indikator awal kemungkinan manipulasi dokumen digital. Meskipun model menunjukkan kinerja tinggi, ruang lingkup penelitian ini masih terbatas pada atribut *font bold* sebagai indikator utama. Pengembangan selanjutnya dapat mencakup eksplorasi atribut *font* lain serta validasi pada dokumen dari dunia nyata.

Kata Kunci : Pemalsuan Dokumen, Dokumen Digital, Citra, CNN

Abstract

Digital document forgery and editing pose significant threats to information security and the authenticity of official records. A recent example is the 2024 case of falsified population documents for school admissions (PPDB), where Ato et al. (Kompas, 2024) reported data manipulation and alterations to documents such as family registration cards (Kartu Keluarga/KK) to exploit the zoning system. This case highlights the vulnerability of digital documents to visual manipulation, particularly through font inconsistencies within their structure. This study aims to automatically detect font inconsistencies in digital documents using a Convolutional Neural Network (CNN) architecture. The model was trained on 100,000 samples from the Document Font Recognition Dataset (DTFR), with preprocessing steps including grayscale conversion, normalization, and resizing of text images to 32×32 pixels. The CNN was designed with two convolutional layers, max pooling, dropout, and dense layers. The evaluation results show an accuracy of 96.85%, with average precision, recall, and F1-score values of 0.97, outperforming a previous SVM-based approach that achieved only

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 8 ; Nomor 2 ; Juli 2025 ; Page 214-226
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

94.6%. These findings demonstrate that CNN is effective for detecting font inconsistencies as an early indicator of potential document manipulation. Although the model achieved high performance, this study is currently limited to detecting bold font attributes. Future work may explore additional font attributes and extend validation to real-world document sources to enhance the model's generalization capability.

Keyword : *Document Forgery, Digital Document, Image, CNN*

1. PENDAHULUAN

Pemalsuan dan penyuntingan dokumen digital menjadi isu yang semakin mengkhawatirkan seiring dengan meningkatnya digitalisasi dalam berbagai sektor, seperti pemerintahan, pendidikan dan keuangan[1]. Dokumen digital yang dipalsukan seperti ijazah, kontrak, atau identitas dapat disalahgunakan untuk kepentingan merugikan[2]. Salah satu kasus nyata terjadi pada proses Penerimaan Peserta Didik Baru (PPDB) tahun 2024, di mana ditemukan praktik pemalsuan dokumen kependudukan seperti Kartu Keluarga (KK) untuk memanipulasi sistem zonasi pendidikan[3]. Kasus ini menunjukkan bahwa dokumen digital sangat rentan terhadap manipulasi. Indikator awal dari manipulasi dokumen adalah ketidakkonsistenan dalam penggunaan *font*, seperti perubahan gaya huruf yang tidak sesuai dengan format atau gaya penulisan asli dokumen [4].

Deteksi manual terhadap ketidakkonsistenan *font* sangat bergantung pada kejelian manusia, dan rawan kesalahan. Sejumlah penelitian telah dilakukan untuk mendeteksi pemalsuan dokumen menggunakan berbagai pendekatan, seperti analisis metadata, deteksi pola tanda tangan, serta pengenalan struktur dokumen. Di sisi lain, klasifikasi *font* dalam dokumen digital telah menjadi topik penting dalam bidang *computer vision* dan *document analysis*, dengan pendekatan berbasis *machine learning* maupun *deep learning* [4]. Namun, sebagian besar studi sebelumnya lebih fokus pada pengenalan *font* untuk keperluan *Optical Character Recognition* (OCR), bukan sebagai indikator penyuntingan atau manipulasi dokumen. Identifikasi perubahan *font* secara manual juga memerlukan waktu dan keahlian khusus sehingga tidak efisien dalam skala besar [5].

Penelitian oleh Alamin et al. menggunakan metode *Support Vector Machine* (SVM) untuk untuk mengklasifikasikan lima jenis *font* populer dalam *grayscale* dan memperoleh akurasi 94,6%, tetapi masih terbatas pada karakter individu dan tergantung pada fitur manual [6]. Sementara itu, *Convolutional Neural Network* (CNN) telah banyak digunakan untuk mendeteksi pemalsuan dokumen digital karena kemampuannya dalam mengekstraksi fitur visual secara otomatis [7]. CNN dipilih karena kemampuannya dalam mengenali pola visual secara otomatis dan akurat[8]. Indriani S. et al. menerapkan CNN untuk verifikasi tanda tangan dengan akurasi tinggi (97%), namun hanya terbatas pada aspek biometrik, bukan font[9]. Sementara itu, Qaroush et al. menggunakan CNN untuk OCR teks Arab dengan akurasi hampir sempurna (99,97%), tetapi fokus penelitiannya adalah pengenalan karakter, bukan deteksi ketidakkonsistenan *font*[10].

Dengan demikian, gap penelitian ini pada pemanfaatan CNN untuk mendeteksi ketidakkonsistenan *font* sebagai indikator pemalsuan dan penyuntingan dokumen digital, berbeda dari OCR biasa atau verifikasi tanda tangan. Penelitian ini menggunakan dataset *Document Font Recognition Dataset* (DTFR) yang berisi berbagai jenis *font* dalam dokumen digital untuk melatih model agar mampu mengenali dan membedakan jenis *font* yang digunakan dalam sebuah dokumen. Adapun pada arsitektur CNN penelitian ini resolusi input diubah ke dalam bentuk yang lebih kecil seperti 32×32 untuk menghasilkan performa klasifikasi yang kompetitif dan mengurangi kebutuhan komputasi secara signifikan sehingga dapat menjaga efisiensi proses pelatihan[11]. Arsitektur CNN yang digunakan dirancang dengan menggunakan dua lapisan konvolusional, masing-masing dengan 32 dan 64 filter berukuran 3×3 dan fungsi aktivasi Rectified Linear Unit (ReLU), yang bertugas mengekstraksi fitur visual dari teks sehingga nilai akhir citra hasil konvolusi mempertahankan pola visual yang penting[12]. Kemudian menggunakan lapisan *max-pooling* yang berfungsi untuk mereduksi dimensi spasial serta mempercepat proses pelatihan[13]. Satu lapisan fully connected (dense) dengan 128 neuron untuk menggabungkan fitur hasil ekstraksi dari lapisan sebelumnya. Lapisan *dropout* sebesar 0,5 diterapkan sebagai teknik regularisasi guna mengurangi risiko *overfitting*[14]. Terakhir, lapisan *output softmax* dengan dua neuron digunakan untuk klasifikasi dua kelas: teks *bold* dan non-*bold*.

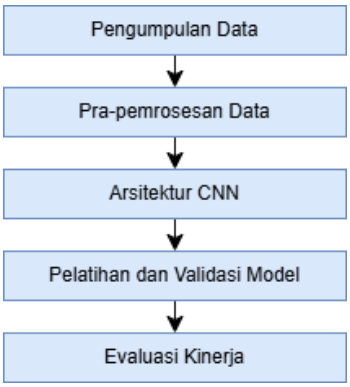
Penelitian ini memiliki kontribusi dalam mendeteksi ketidakkonsistenan *font* berbasis citra dokumen. Berbeda dari pendekatan OCR umum, fokus penelitian ini adalah pada deteksi perubahan gaya penulisan *font*, khususnya atribut *bold*, sebagai indikator awal penyuntingan atau pemalsuan. Selain itu, penelitian ini membandingkan performa CNN dengan pendekatan SVM, serta mengevaluasi model pada dataset besar berjumlah 100.000 sampel untuk menguji akurasi dan generalisasi. Tujuan dari penelitian ini adalah melakukan evaluasi dan pengembangan model deteksi otomatis berbasis CNN yang mampu mengidentifikasi ketidakkonsistenan *font* secara akurat dalam dokumen digital.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Pendekatan berbasis *Convolutional Neural Network* (CNN) digunakan untuk mendeteksi ketidakkonsistenan *font* sebagai indikator pemalsuan dan penyuntingan dalam dokumen digital. Penelitian akan menggunakan metode berikut:

1. Pengumpulan Data
   Mengumpulkan dataset dokumen digital yang terdiri dari dokumen asli dan dokumen yang telah dimanipulasi. Dataset akan mencakup variasi *font*, gaya penulisan dan teknik penyuntingan yang berbeda-beda.
2. Pra-pemrosesan Data
   Melakukan segmentasi dan ekstraksi area teks dari dokumen digital dan mengubah citra teks menjadi format yang sesuai untuk pemrosesan CNN.
3. Arsitektur CNN
   Merancang arsitektur CNN yang terdiri dari lapisan konvolusional, lapisan *pooling*, dan lapisan *fully connected*. Kemudian peningkatan akurasi dengan mengoptimalkan *hyperparameter* CNN seperti jumlah lapisan, ukuran filter dan fungsi aktivasi.
4. Pelatihan dan Validasi Model
   Membagi dataset menjadi set pelatihan dan validasi untuk memastikan generalisasi model dan menerapkan teknik augmentasi data untuk meningkatkan keragaman data pelatihan dan mengatasi *overfitting*.
5. Evaluasi Kinerja
   Menguji model pada set data uji yang terpisah untuk mengevaluasi akurasi kinerja model CNN pada deteksi pemalsuan dokumen.



Gambar 1. Tahapan penelitian

2.2 Dataset

Dalam penelitian ini digunakan *Document Font Recognition Dataset* (DTFR), yaitu dataset publik yang tersedia di Kaggle dan terdiri atas 31.100 file dokumen digital dalam format gambar (.png) dengan label dalam format JSON yang memuat daftar blok teks (*lines*) yang dipecah menjadi potongan-potongan (*spans*). Format JSON berisi informasi gaya tulisan seperti *bold*, *italic*, *monospaced* dan *serif* dalam format *boolean (true/false)*. Label dalam dataset ini tidak mencantumkan nama font eksplisit, melainkan berfokus pada karakteristik visual dari teks. Berdasarkan hasil *parsing* seluruh struktur label, diketahui bahwa dataset ini mengandung total sekitar 1.976.600 span teks. Namun, untuk keperluan pelatihan dan pengujian model, digunakan sebanyak 100.000 span teks yang dikumpulkan dari 3.804 dari total 31.100 dokumen JSON (sekitar 12% dari total dokumen), hingga mencapai batas maksimum (max_data=100_000) yang telah ditentukan dalam program ekstraksi.

Dari data tersebut, didapatkan distribusi label yaitu 49.982 span teks berlabel *bold* (49,98%) dan 50.018 span berlabel non-*bold* (50,02%). Distribusi label pada data yang digunakan sudah alami seimbang sehingga proses penyeimbangan data tidak diperlukan dalam penelitian ini. Dari total 100.000 span teks hasil ekstraksi, selisih data berlabel *bold* dan non-*bold* sangat kecil yaitu 0,036%[15]. Dengan distribusi hampir 50:50 menunjukkan bahwa tidak perlu untuk menerapkan teknik *balancing* seperti *undersampling* atau *oversampling*. Selanjutnya, data dibagi menjadi 80.000 untuk pelatihan dan 20.000 untuk pengujian, dan seluruh proses ekstraksi serta pra-pemrosesan dilakukan menggunakan bahasa pemrograman Python.

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 8 ; Nomor 2 ; Juli 2025 ; Page 214-226
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

Gambar 2. Sampel Dataset

2.3 Pra-pemrosesan Data

Langkah pra-pemrosesan dilakukan untuk mengekstrak informasi yang relevan dari dataset yang digunakan, pada DTFR ini file .json dibaca untuk mengidentifikasi area teks pada dokumen dan label gaya *font*-nya (khususnya atribut *bold* sebagai indikator ketidakkonsistenan). Setiap area teks yang berlabel dihubungkan dengan citra dokumen yang sesuai. Citra teks dipotong berdasarkan koordinat *bounding box* (bbox) untuk membatasi area citra yang dideteksi[16]. Hasil ekstraksi citra dikonversi ke dalam format *grayscale* untuk mengurangi kompleksitas data visual dari tiga kanal warna *Red, Green, Blue* (RGB) tanpa menghilangkan informasi penting seperti bentuk, tepi, dan ketebalan huruf. Hal ini dilakukan untuk menyederhanakan proses komputasi dan mempercepat pelatihan CNN, juga membantu mencegah *overfitting* akibat informasi warna yang tidak relevan dalam tugas klasifikasi font. Efektivitas penggunaan *grayscale* dalam klasifikasi visual berbasis CNN telah dibuktikan dalam penelitian oleh Wang & Lee, yang menunjukkan bahwa fitur struktural tetap dapat dikenali dengan baik meskipun hanya menggunakan kanal *grayscale*[17].

Setiap gambar diubah ukurannya menjadi 32×32 piksel untuk menghasilkan performa klasifikasi yang efisien dengan kebutuhan komputasi yang lebih rendah. Kemudian data dinormalisasi ke rentang [0,1] untuk diproses oleh CNN. Proses ini menghasilkan dua array yaitu X sebagai kumpulan gambar teks, dan y sebagai label binari (0 untuk non-*bold*, 1 untuk *bold*). Penelitian ini secara khusus memfokuskan klasifikasi pada atribut *bold* sebagai indikator utama ketidakkonsistenan *font*, karena perubahan ketebalan huruf merupakan salah satu bentuk manipulasi dokumen digital yang paling umum dan paling mudah terdeteksi dalam dokumen palsu seperti yang dilakukan oleh Chernyshova et al. menggunakan CNN untuk mendeteksi ketidaksesuaian *font* dalam dokumen identitas yang diambil menggunakan kamera *smartphone*[18]. Atribut lain seperti *italic*, *monospaced*, dan *serif* memang juga tersedia dalam metadata DTFR, namun tidak dijadikan fokus dalam tahap klasifikasi model ini karena keterbatasan ruang lingkup dan untuk menjaga spesifikasi tujuan penelitian.

2.4 Arsitektur CNN

Model CNN dirancang secara bertahap untuk mengekstraksi fitur visual dari citra teks hasil potongan dokumen. Arsitektur terdiri dari dua lapisan konvolusional dengan jumlah filter masing-masing 32 dan 64, ukuran kernel 3×3 dan fungsi aktivasi ReLU. Lapisan ini bertugas mengekstraksi pola visual penting dari teks seperti ketebalan huruf dan batas tepi, yang menjadi karakteristik utama dalam membedakan teks *bold* dan non-*bold*. Setelah setiap lapisan konvolusi, digunakan lapisan *MaxPooling* 2×2 untuk mereduksi dimensi spasial dari citra, mempercepat proses pelatihan, dan meningkatkan ketahanan model terhadap translasi minor. Kemudian, diterapkan *Dropout layer* sebesar 0,5 sebagai teknik regularisasi untuk menghindari *overfitting* dengan cara mengabaikan sebagian neuron secara acak saat pelatihan. Selanjutnya, hasil ekstraksi fitur diratakan menggunakan lapisan *Flatten*, lalu dimasukkan ke dalam lapisan *fully connected (dense)* dengan 128 neuron, yang menggabungkan semua fitur untuk klasifikasi akhir. Lapisan terakhir adalah *output layer* dengan dua neuron dan fungsi aktivasi *softmax*, yang digunakan untuk mengklasifikasikan teks ke dalam dua kelas yaitu *bold* dan non-*bold*. Pemilihan *softmax* dilakukan untuk menghasilkan output probabilitas dari masing-masing kelas sehingga lebih sesuai untuk klasifikasi dua kelas dengan evaluasi berbasis metrik *precision*, *recall* dan *f1-score*[19].

2.5 Pelatihan dan Validasi Model

Dataset dibagi menjadi 80% data latih (80.000 sampel) dan 20% data uji (20.000 sampel). Model CNN dilatih selama lima *epoch*, menggunakan *optimizer Adam* dan *loss function categorical cross-entropy*. Arsitektur model menggunakan dua lapisan konvolusional dan total lima lapisan utama, termasuk *max pooling*, *dropout*, dan *dense layer*. Teknik data augmentasi ringan juga digunakan, seperti rotasi acak dalam kisaran ±10 derajat dan *flipping* horizontal untuk meningkatkan keragaman data pelatihan dan mencegah *overfitting*[20]. Label yang digunakan bersifat biner, yaitu 1 untuk teks *bold* dan 0 untuk non-*bold*, berdasarkan atribut *bold* pada metadata file JSON. Meskipun informasi lain seperti *italic*,

serif dan *monospaced* tersedia, penelitian ini hanya memfokuskan klasifikasi pada atribut *bold* sebagai indikator utama ketidakkonsistenan *font*.

2.6 Evaluasi Kinerja

Setelah model dilatih, langkah selanjutnya adalah melakukan evaluasi dengan menggunakan beberapa metrik, yaitu akurasi, presisi, recall, dan F1-Score. Akurasi digunakan untuk mengindikasikan seberapa baik model dapat mengklasifikasikan data secara keseluruhan dengan benar, sementara presisi menunjukkan proporsi prediksi positif yang benar. Recall mengukur kemampuan model dalam mendeteksi kelas positif dengan akurat, sedangkan F1-Score adalah kombinasi dari presisi dan recall. F1-Score yang tinggi menunjukkan bahwa model memiliki keseimbangan yang baik antara *Precision* dan *Recall*[21]. Pengukuran performa klasifikasi data model menggunakan metrik evaluasi dalam klasifikasikan data tersebut[22]. Selain itu, digunakan *confusion matrix* untuk melihat distribusi prediksi antar kelas, serta dilakukan perbandingan dengan model pembanding berdasarkan akurasi akhir[23]. Evaluasi dilakukan pada 20% data uji yang telah dipisahkan secara acak dari data utama. Perhitungan ini penting untuk memahami bagaimana model menangani prediksi benar dan salah pada masing-masing kategori, serta untuk mengetahui keseimbangan kinerjanya terhadap seluruh data[24].

Implementasi model dilakukan menggunakan bahasa pemrograman Python 3.11.13, dengan pustaka TensorFlow 2.18.0 dan Keras 3.8.0 sebagai *framework* utama pengembangan CNN. Eksperimen dijalankan di lingkungan Google Colaboratory tanpa akselerasi GPU (CPU-only). Pemrosesan dan pelatihan data didukung oleh pustaka tambahan seperti, NumPy untuk manipulasi array dan normalisasi data, OpenCV untuk pembacaan gambar, konversi ke *grayscale*, *cropping* dan *resize* citra, tqdm untuk menampilkan progres saat pemrosesan batch data, matplotlib.pyplot untuk visualisasi grafik pelatihan dan metrik evaluasi, scikit-learn untuk pembagian data latih-uji dan evaluasi metrik model, serta pustaka standar os dan json untuk navigasi file sistem dan membaca label dari file JSON.

3. HASIL DAN PEMBAHASAN

Penelitian ini dilakukan melalui serangkaian tahapan eksperimental untuk membangun sistem deteksi ketidakkonsistenan *font* sebagai indikator awal pemalsuan dokumen digital. Tahapan implementasi dimulai dari pengumpulan dan pra-pemrosesan data, perancangan arsitektur CNN, hingga pelatihan dan evaluasi model.

3.1 Hasil Pra-Pemrosesan Data

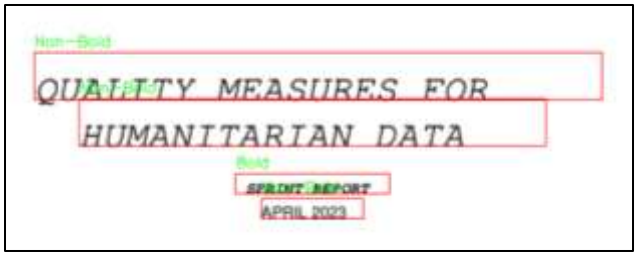
Dataset DTFR diproses dengan mengekstraksi area teks dari citra dokumen berdasarkan informasi koordinat yang tersedia pada file .json. Fokus utama adalah membedakan area teks berjenis *bold* dan non-*bold* sebagai indikator awal dari ketidakkonsistenan *font*. Tahap pra-pemrosesan merupakan langkah krusial untuk menyiapkan data agar dapat digunakan dalam pelatihan model CNN. Pada dataset ini yang berisi dokumen digital dalam bentuk gambar (.png) dan label terkait dalam format .json. Setiap file .json berisi informasi struktur dokumen, termasuk koordinat *bounding box* (bbox) dari area teks serta atribut *font* seperti ukuran huruf, tebal (*bold*), miring (*italic*), jenis huruf (*serifed*, *monospaced* dan jenis huruf lainnya). Penelitian ini hanya fokus pada atribut "*bold*", karena ketidakkonsistenan tebal huruf dapat menjadi salah satu indikator terjadinya manipulasi dokumen. Langkah-langkah pra-pemrosesan data :

a. Pembacaan file JSON dan gambar
   Setiap file .json di-*parsing* untuk mendapatkan posisi area teks (*bounding box*) dari setiap baris atau span teks, gambar terkait dibaca dalam mode *grayscale* menggunakan OpenCV (*cv2.IMREAD_GRAYSCALE*) agar lebih ringan diproses. Berikut ini hasil pembacaan citra dokumen dalam mode *grayscale* :

Gambar 3. Citra Dokumen Dalam Mode *Grayscale*

   Hasil awal dari proses membaca file JSON dan gambar dokumen dapat dilihat pada gambar di bawah ini. File JSON digunakan untuk mendapatkan posisi area teks (*bounding box*) di dalam gambar dokumen. Visualisasi yang digunakan yaitu kotak-kotak (*bounding box*) yang mengelilingi tiap baris atau span teks pada citra dokumen. Fungsi utama tahap ini adalah mendeteksi lokasi teks sebelum dilakukan pemrosesan lebih lanjut.

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 8 ; Nomor 2 ; Juli 2025 ; Page 214-226
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

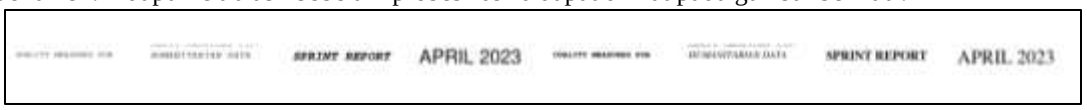


Gambar 4. Hasil Pembacaan File JSON dan Gambar Untuk *Bounding Box*

b. Ekstraksi dan pemotongan area teks
Berdasarkan koordinat *bounding box*, sistem memotong bagian citra yang mengandung teks dari keseluruhan gambar dokumen. Gambar berikut menampilkan potongan-potongan citra hasil ekstraksi area teks berdasarkan koordinat bounding box yang telah didapat sebelumnya. Setiap potongan hanya berisi bagian gambar yang relevan dengan teks, sehingga informasi di luar teks dihilangkan. Tujuannya adalah agar model hanya fokus pada fitur visual dari teks.

Gambar 5. Hasil Pemotongan Area Teks

c. *Resize* gambar
Setiap potongan teks hasil ekstraksi diubah ukurannya menjadi 32x32 piksel untuk menyeragamkan dimensi input ke CNN *(cv2.resize)*. Citra ini masih memiliki ukuran bervariasi sesuai ukuran teks aslinya dalam dokumen. Adapun citra asli sebelum proses *resize* dapat dilihat pada gambar berikut :



Gambar 6. Citra Asli Sebelum *Resize*

Resize dilakukan untuk menyeragamkan ukuran input ke model CNN sehingga proses pelatihan menjadi lebih konsisten dan efisien. Berikut hasil dari perbandingan citra asli dokumen dengan hasil resize yang telah dilakukan.

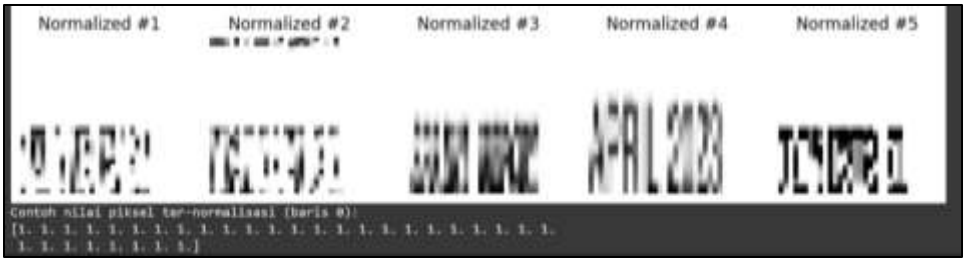
Gambar 7. Perbandingan Hasil Citra Asli dan *Resize*

d. Pemberian label
Gambar berikut menampilkan potongan-potongan citra yang sudah diberi label. Label 1 (*Bold)* untuk area teks yang dalam atribut JSON-nya memiliki "*bold*" : *True* dan label 0 (Non-*Bold*) untuk area teks yang tidak memiliki atribut "*bold*": *True*. Label ini digunakan sebagai target dalam pelatihan model untuk mendeteksi teks tersebut *bold* atau tidak..

Gambar 8. Hasil Pra-Pemrosesan Dengan Label

e. Normalisasi data:
Setiap nilai piksel dalam gambar dinormalisasi ke rentang [0, 1] agar stabil dalam pelatihan model.

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 8 ; Nomor 2 ; Juli 2025 ; Page 214-226
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index



Gambar 9. Hasil Normalisasi Data

f. Penyimpanan dalam format .npy:
Untuk efisiensi pemuatan saat pelatihan, seluruh dataset hasil pra-pemrosesan disimpan dalam format NumPy (X 100k.npy untuk gambar, dan y100k.npy untuk label). Untuk menghemat penggunaan memori (RAM) selama proses pelatihan di Google Colab, dari seluruh data yang tersedia dalam dataset DTFR, hanya 100.000 sampel data yang digunakan sebagai subset. Pemilihan ini dilakukan secara berurutan dari data yang berhasil diproses, dan tetap mempertahankan distribusi label yang seimbang. Hasil akhir pra-pemrosesan menghasilkan :

Tabel 1. Hasil Pra-Pemrosesan Data

Hasil Akhir	
Jumlah Total Citra Teks	100.000
Label *Bold* (1)	49.982
Label Non-*Bold* (0)	50.018

Distribusi label yang hampir seimbang ini memastikan model dapat belajar secara adil tanpa bias ke salah satu kelas. Proses pra-pemrosesan ini menjadi fondasi penting dalam mendukung pelatihan model *deep learning* yang efektif dan efisien, terutama saat sumber daya komputasi terbatas.

3.2 Hasil Pelatihan CNN

Arsitektur CNN dirancang untuk mendeteksi ketidakkonsistenan *font*, khususnya membedakan antara teks *bold* dan non-*bold* sebagai indikasi pemalsuan dokumen. Arsitektur model terdiri atas:

a. 2 lapisan konvolusional, masing-masing dengan 32 dan 64 filter berukuran 3x3, dilengkapi dengan fungsi aktivasi ReLU untuk mengekstraksi fitur visual dari teks sehingga nilai akhir citra hasil konvolusi bernilai sama dengan konvolusinya.
b. 2 lapisan *max-pooling* 2x2 untuk mereduksi dimensi spasial dan mempercepat proses pelatihan.
c. 1 lapisan *fully connected* (*dense*) dengan 128 neuron untuk menggabungkan fitur-fitur hasil konvolusi.
d. Lapisan *Dropout* sebesar 0.5 digunakan sebagai teknik regularisasi guna menghindari *overfitting*.
e. Lapisan output *softmax* dengan 2 neuron (klasifikasi 2 kelas: *bold* dan non-*bold*).

Model dilatih menggunakan data hasil pra-pemrosesan sebanyak 80.000 data latih dan diuji pada 20.000 data uji, dengan *batch size* 128 selama 5 *epoch*. Hasil pelatihan yang terlihat pada tabel menunjukkan tren akurasi yang meningkat dan *loss* yang menurun seiring bertambahnya *epoch*.

Tabel 2. Tren Akurasi Hasil Pelatihan CNN

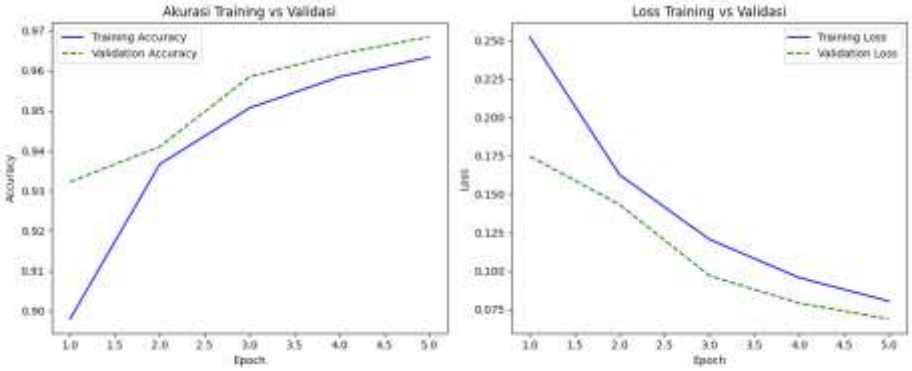
***Epoch* ke -**	**Akurasi**	***Loss***
1	0,8487	0,3390
2	0,9333	0,1733
3	0,9473	0,1278
4	0,9582	0,0985
5	0,9627	0,0836
Hasil	**0,9685**	**0,0315**

Akurasi pada data uji setelah pelatihan adalah 96,85%, menunjukkan kemampuan model untuk melakukan generalisasi yang sangat baik terhadap data baru. Hasil ini memvalidasi bahwa arsitektur CNN yang digunakan sangat efektif pada klasifikasi teks dokumen berdasarkan jenis *font* dan dapat dimanfaatkan untuk mendeteksi indikasi pemalsuan berbasis visual *font*.

3.3 Visualisasi Proses Pelatihan

Proses pelatihan model CNN divisualisasikan dalam dua grafik utama yaitu grafik akurasi dan grafik *loss*, yang masing-masing membandingkan performa pada data pelatihan dan data validasi selama lima *epoch*.

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 8 ; Nomor 2 ; Juli 2025 ; Page 214-226
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

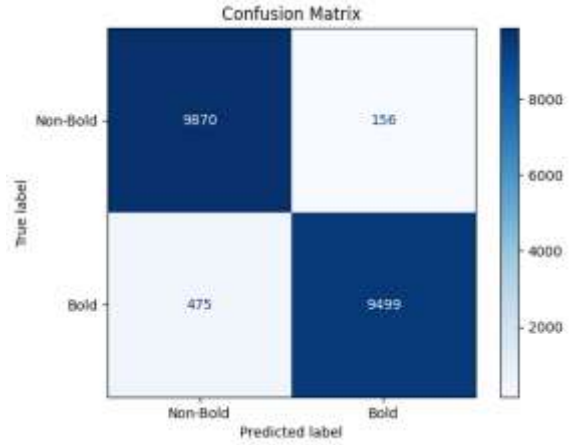


Gambar 10. Grafik Akurasi dan *Loss Training*

Pada grafik sebelah kiri, terlihat bahwa akurasi data pelatihan (*Training Accuracy*, garis biru) meningkat dari sekitar 89% di *epoch* 1 menjadi hampir 96% di *epoch* 5. Akurasi validasi (*Validation Accuracy*, garis hijau putus-putus) menunjukkan pola yang serupa, naik dari sekitar 92% hingga mencapai 96,5% di akhir pelatihan. Perbedaan akurasi antara *training* dan validasi sangat kecil, mengindikasikan bahwa generalisasi model baik dan tidak terjadi *overfitting*. Sedangkan pada grafik di sebelah kanan nilai *loss* pada data pelatihan (*Training Loss*, garis biru) menurun secara signifikan dari sekitar 0.26 di *epoch* 1 menjadi 0.09 di *epoch* 5. Nilai *loss* validasi (*Validation Loss*, garis hijau putus-putus) juga menurun secara konsisten dari 0.19 ke 0.0797, mengikuti tren yang sama dengan *training loss*. Konsistensi penurunan *loss* di grafik yang tersedia dapat terlihat model data terlatih dengan efektif, tidak mengalami masalah *underfitting* maupun *overfitting*. Visualisasi ini menjelaskan bahwa model CNN yang dibangun belajar secara stabil dan efisien dari data. Akurasi yang terus meningkat dan *loss* yang menurun di kedua set (*training* dan validasi) menunjukkan bahwa arsitektur model dan parameter yang digunakan sudah optimal untuk tugas deteksi *font* dalam dokumen digital.

3.4 Evaluasi Model

Model dievaluasi menggunakan *Confusion Matrix* dan *Classification Report* untuk mengukur performa dalam mengklasifikasikan teks menjadi dua kelas: *Bold* dan Non-*Bold*. *Confusion Matrix* dimanfaatkan untuk menilai kinerja model dengan menghitung berbagai metrik evaluasi, seperti akurasi, presisi, *Recall*, dan skor F1[23].



Gambar 11. *Confusion Matrix*

Dari *Confusion Matrix*, kita dapat melihat distribusi prediksi model terhadap label yang sebenarnya dengan hasil berikut :

a. *True Positives* (*Bold* terhadap *Bold*), jumlah data berlabel "*Bold*" dan berhasil diprediksi dengan benar sebagai "*Bold*" oleh model adalah 9499.
b. *True Negatives* (Non-*Bold* terhadap Non-*Bold*), jumlah data berlabel "Non-*Bold*" dan berhasil diprediksi dengan benar sebagai "Non-*Bold*" oleh model adalah 9870.
c. False *Positives* (Non-*Bold* terhadap *Bold*), jumlah data berlabel "Non-*Bold*" dan diprediksi salah sebagai "*Bold*" oleh model adalah 156.
d. False *Negatives* (*Bold* terhadap Non-*Bold*), jumlah data berlabel "*Bold*" dan diprediksi salah sebagai "Non-*Bold*" oleh model adalah 475.

Berdasarkan hasil *Confusion Matrix* tersebut, *False Positive* yang berarti teks yang sebenarnya sah (non-*bold*) diprediksi sebagai *bold*, dapat menyebabkan *false* alarm dalam sistem deteksi dokumen. Sebaliknya, *false negative* (FN)

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 8 ; Nomor 2 ; Juli 2025 ; Page 214-226
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

berpotensi lebih berbahaya, karena bagian yang seharusnya terdeteksi sebagai manipulasi (*bold*) justru luput dari deteksi. Oleh karena itu, *recall* yang tinggi menunjukkan dapat meminimalisir kelolosan pemalsuan data. Selanjutnya dilakukan penilaian performa model dalam mengklasifikasikan teks *bold* dan non-*bold*, dilakukan analisis *Confusion Matrix*. Dan dihitung metrik evaluasi utama *Classification Report* seperti *Accuracy*, *Precision*, *Recall*, dan *f1-score* untuk masing-masing kelas. Perhitungan ini penting untuk memahami bagaimana model menangani prediksi benar dan salah pada masing-masing kategori, serta untuk mengetahui keseimbangan kinerjanya terhadap seluruh data.

a. Akurasi
Akurasi mengukur proporsi prediksi yang benar dibandingkan dengan seluruh jumlah data.

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

$$\text{Akurasi} = \frac{9870 + 9499}{9870 + 9499 + 156 + 475} = \frac{19369}{20000} = 0{,}96845$$

b. *Precision*
Precision mengukur seberapa banyak prediksi yang benar-benar positif

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

1. *Precision* untuk kelas *Bold*
$$Precision = \frac{9499}{9499 + 156} = \frac{9499}{9655} \approx 0.9840$$

2. *Precision* untuk kelas Non-*Bold*
$$Precision = \frac{9870}{9870 + 475} = \frac{9870}{10345} \approx 0.9541$$

c. *Recall*
Recall mengukur seberapa banyak data positif yang berhasil dikenali.

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

1. *Recall* untuk kelas *Bold*
$$Recall = \frac{9499}{9499 + 475} = \frac{9499}{9974} \approx 0.9524$$

2. *Recall* untuk kelas Non-*Bold*
$$Recall = \frac{9870}{9870 + 156} = \frac{9870}{10026} \approx 0.9844$$

d. *F1-score*
F1-score adalah rata-rata keselerasan dari *Precision* dan *Recall*..

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

1. *F1-score* untuk kelas *Bold*
$$F1 - Score = 2 \times \frac{0.9840 \times 0.9524}{0.9840 + 0.9524} \approx 2 \times \frac{0.9371}{1.9364} \approx 0.967$$

2. *F1-score* untuk kelas Non-*Bold*
$$F1 - Score = 2 \times \frac{0.9541 \times 0.9844}{0.9541 + 0.9844} \approx 2 \times \frac{0.9393}{1.9385} \approx 0.968$$

Tabel 3. Hasil Evaluasi *Precision*, *Recall* dan *F1-score*

Kelas	*Precision*	*Recall*	*F1-score*	Support
Bold	0,98	0,95	0,97	9.974
Non-*Bold*	0,95	0,98	0,97	10.026
Rata-rata	0,97	0,97	0,97	20.000

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 8 ; Nomor 2 ; Juli 2025 ; Page 214-226
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

Rata-rata Tertimbang	0,97	0,97	0,97	20.000
Hasil			**0,97 (97%)**	**20.000**

Berdasarkan tabel di atas terlihat bahwa *Precision* yang tinggi untuk kedua kelas menunjukkan bahwa model jarang memberikan prediksi positif palsu. *Recall* yang juga tinggi memastikan bahwa klasifikasi sebagian besar sebagai "*Bold*" atau "Non-*Bold*" telah berhasil dikenali. *F1-score* sebesar 0.96–0.97 menunjukkan keselarasan yang sangat baik antara *Precision* dan *Recall*. Akurasi keseluruhan mencapai 97%, yang berarti hanya 3% kesalahan pada dataset pengujian sebanyak 20.000 sampel.

3.5 Uji Coba Pada Dokumen Nyata

Pada bagian ini, menjelaskan model CNN yang telah dilatih diuji terhadap dokumen asli dan dokumen hasil penyuntingan (pemalsuan) yang diunggah secara manual. Sepasang dokumen Badan Akreditasi Nasional Pendidikan Anak Usia Dini dan Pendidikan Nonformal (BAN PAUD dan PNF) yang digunakan terdiri dari versi asli (sampel_undangan_rapat.png) sesuai dengan file yang diberikan oleh BAN PAUD dan PNF mengenai pemberitahuan Rapat Koordinasi Nasional dan versi hasil penyuntingan (forgery_undangan_rapat.png) yang sudah diedit pada nomor telepon "7658424" menjadi "7558442", Meeting ID "857-105-3278" menjadi "587-501-2378" dan Password "BANP#2021" menjadi "BANP*2021". Model CNN yang telah dilatih digunakan untuk memindai kedua dokumen dan mengklasifikasikan setiap span teks berdasarkan ketebalan huruf (*bold* atau non-*bold*). Hasil pemindaian menunjukkan bahwa terdapat sejumlah perbedaan klasifikasi gaya font yang mengindikasikan adanya manipulasi pada bagian tertentu dokumen hasil penyuntingan. Eksperimen ini bertujuan untuk mengevaluasi kemampuan model dalam mendeteksi perubahan gaya font (*bold* vs non-*bold*) sebagai indikasi pemalsuan dan penyuntingan dokumen digital.

Gambar 12. Dokumen Asli

Gambar 13. Hasil Deteksi Font Dokumen Asli

Pada gambar 12. menampilkan dokumen yang belum mengalami perubahan atau manipulasi. Dokumen asli digunakan sebagai acuan dalam proses deteksi pemalsuan. Semua elemen teks pada dokumen ini konsisten baik dari segi font, ukuran, maupun atribut lainnya. Kemudian pada gambar 13. Setelah dilakukan pra-pemrosesan data menunjukkan hasil proses deteksi font pada dokumen asli. Sistem mendeteksi dan memetakan area-area teks dalam dokumen, memastikan bahwa seluruh font yang digunakan seragam dan sesuai dengan standar dokumen asli. Hasil ini menjadi pembanding untuk mendeteksi adanya anomali pada dokumen yang dicurigai palsu.

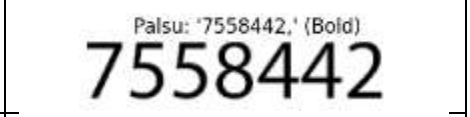
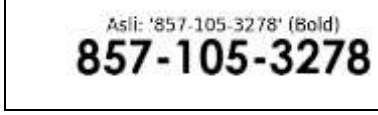
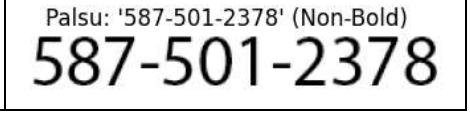
Gambar 14. Dokumen Pemalsuan

Gambar 15. Hasil Deteksi Perubahan Font Dokumen

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 8 ; Nomor 2 ; Juli 2025 ; Page 214-226
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

Sedangkan pada gambar selanjutnya memperlihatkan contoh dokumen yang telah dimanipulasi atau dipalsukan. Pada dokumen ini terdapat perbedaan atau ketidakkonsistenan pada elemen font, yang mungkin terjadi akibat penyuntingan sebagian teks. Perubahan ini bisa menjadi indikator adanya upaya memalsukan isi dokumen. Gambar 15. menampilkan hasil deteksi sistem terhadap dokumen palsu, khususnya pada bagian font. Sistem mendeteksi area teks yang font-nya berbeda dari dokumen asli dan menandai bagian tersebut. Hasil ini membantu dalam mengidentifikasi bagian dokumen yang mengalami manipulasi font, sehingga memudahkan proses verifikasi keaslian dokumen. Berikut hasil perubahan font yang terdeteksi setelah dilakukan proses *cropping* dan *resize*.

Tabel 4. Perubahan Font Terdeteksi

Dokumen Asli	Dokumen Pemalsuan
	
	

Berdasarkan tabel di atas menunjukkan bahwa pemalsuan dan penyuntingan dokumen digital dapat dideteksi menggunakan CNN, meskipun ada beberapa kata yang tidak termasuk dalam manipulasi dan ada kata yang tidak terdeteksi manipulasi, hal ini menunjukkan bahwa perlu dilakukan lagi pelatihan CNN pada model baru yang akan dideteksi untuk mempelajari model dokumen asli sehingga perubahan *font* terdeteksi dapat lebih baik dan akurat.

4. KESIMPULAN

Penelitian ini berhasil mengembangkan dan mengevaluasi model deteksi otomatis terhadap ketidakkonsistenan jenis *font* dalam dokumen digital, yang dapat menjadi indikator awal adanya penyuntingan atau pemalsuan. Dengan menggunakan pendekatan *Convolutional Neural Network* (CNN), model dilatih pada 100.000 sampel dari dataset *Document Font Recognition Dataset* (DTFR), yang telah diproses dalam bentuk citra *grayscale* berukuran 32×32 piksel. Arsitektur CNN terdiri dari dua lapisan konvolusional, *max pooling*, *dropout* dan *dense layer*. Klasifikasi dua kelas untuk yaitu *bold* dan non-*bold* menggunakan *softmax*. Model menunjukkan performa yang sangat baik dengan akurasi sebesar 96,85% dan nilai *precision*, *recall*, serta *f1-score* rata-rata sebesar 0,97, melampaui metode SVM dari penelitian sebelumnya yang mencatat akurasi 94,6%. Selain itu, model diuji pada dokumen digital nyata dan berhasil mengidentifikasi beberapa perubahan gaya font antara versi asli dan versi yang telah dimanipulasi. Meskipun memberikan hasil yang cukup baik, penelitian ini masih terbatas pada deteksi atribut bold sebagai indikator utama. Pengembangan lebih lanjut dapat dilakukan dengan memperluas cakupan ke atribut font lainnya seperti *italic*, ukuran huruf dan jenis *font* tertentu. Selain itu, validasi pada dokumen dari berbagai institusi dan format yang lebih kompleks dapat meningkatkan generalisasi model untuk digunakan dalam konteks nyata yang lebih luas.

UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala atas segala rahmat, nikmat, dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan artikel ini dengan baik. Ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada kedua orang tua tercinta atas doa, dukungan, dan semangat yang tiada henti. Penulis juga menyampaikan rasa hormat dan terima kasih yang mendalam kepada Ibu Prof. Dr. Rika Rosnelly, S.Kom., M.Kom. dan Bapak Dr. Adil Setiawan, S.Kom., M.Kom. selaku dosen pembimbing yang telah memberikan bimbingan, arahan, dan masukan berharga selama proses penelitian. Terima kasih kepada seluruh pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam proses penyelesaian karya ini. Semoga segala kebaikan yang telah diberikan menjadi amal jariyah dan mendapat balasan dari Allah Subhanahu wa Ta'ala.

DAFTAR PUSTAKA

[1] A.-S. Roman, B. Genge, A.-V. Duka, and P. Haller, "Privacy-Preserving Tampering Detection in Automotive Systems," *Electronics*, vol. 10, no. 24, p. 3161, 2021, doi: 10.3390/electronics10243161.

[2] C. Johnson, R. Davies, and M. Reddy, "Using digital forensics in higher education to detect academic misconduct," *Int. J. Educ. Integr.*, vol. 18, no. 1, pp. 1–19, 2022, doi: 10.1007/s40979-022-00104-1.

[3] S. Ato, A. Insan, and A. Diveranta, "Kemendagri Ancam Pidanakan Pemalsu Dokumen Kependudukan untuk Akali PPDB," *Kompas.id*, Jun. 25, 2024. [Online]. Available: https://www.kompas.id/baca/investigasi/2024/06/25/kemendagri-pemalsu-dokumen-kependudukan-bisa-dipidana

[4] J. Koponen, K. Haataja, and P. Toivanen, "Recent Advancements in Machine Vision Methods for Product Code Recognition: A Systematic Review," *F1000research*, vol. 11, p. 1099, 2022, doi: 10.12688/f1000research.124796.1.

[5] Z. Wei and X. Zhang, "Feature Extraction and Retrieval of Ecommerce Product Images Based on Image Processing," *Trait. Du Signal*, vol. 38, no. 1, pp. 181–190, 2021, doi: 10.18280/ts.380119.

[6] Z. Alamin, S. Mutmainah, and M. Hayun, "Optimasi Ekstraksi Fitur Citra Karakter Font Menggunakan Algoritma Support Vector Machines (SVM) untuk Klasifikasi Tipografi", doi: 10.34304/scientific.v2i1.344.

[7] S. Longari, D. H. N. Valcarcel, M. Zago, M. Carminati, and S. Zanero, "CANnolo: An Anomaly Detection System Based on LSTM Autoencoders for Controller Area Network," *Ieee Trans. Netw. Serv. Manag.*, vol. 18, no. 2, pp. 1913–1924, 2021, doi: 10.1109/tnsm.2020.3038991.

[8] R. Hamzehyan, F. Razzazi, and A. Behrad, "Printer Source Identification by Feature Modeling in the Total Variable Printer Space," *J. Forensic Sci.*, vol. 66, no. 6, pp. 2261–2273, 2021, doi: 10.1111/1556-4029.14822.

[9] D. D. Indriani.S, E. J. A. Sinaga, G. Oktavia, H. Syahputra, and F. Ramadhani, "Identifikasi Tanda Tangan Dengan Menggunakan Metode Convolution Neural Network (CNN)," *J-Intech*, vol. 12, no. 1, pp. 138–147, 2024, doi: 10.32664/j-intech.v12i1.1273.

[10] A. Qaroush, A. Awad, M. Modallal, and M. Ziq, "Segmentation-based, omnifont printed Arabic character recognition without font identification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3025–3039, 2022, doi: 10.1016/j.jksuci.2020.10.001.

[11] V. Thambawita, I. Strümke, S. A. Hicks, P. Halvorsen, S. Parasa, and M. A. Riegler, "Impact of image resolution on deep learning performance in endoscopy image classification: An experimental study using a large dataset of endoscopic images," *Diagnostics*, vol. 11, no. 12, 2021, doi: 10.3390/diagnostics11122183.

[12] H. Herdianto and D. Nasution, "Implementasi Metode Cnn Untuk Klasifikasi Objek," *METHOMIKA J. Manaj. Inform. dan Komputerisasi Akunt.*, vol. 7, no. 1, pp. 54–60, 2023, doi: 10.46880/jmika.vol7no1.pp54-60.

[13] P. Meliuwati and E. Kurniati, "Ekstraksi Data Digital Menggunakan Teknik Max Pooling dan Average Pooling," *J. Ris. Mat.*, pp. 137–144, 2022, doi: 10.29313/jrm.v2i2.1338.

[14] X. Liang *et al.*, "R-Drop: Regularized Dropout for Neural Networks," *Adv. Neural Inf. Process. Syst.*, vol. 13, no. NeurIPS, pp. 10890–10905, 2021.

[15] R. C. Moore, D. P. W. Ellis, E. Fonseca, S. Hershey, A. Jansen, and M. Plakal, "Dataset Balancing Can Hurt Model Performance," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2023-June, 2023, doi: 10.1109/ICASSP49357.2023.10095255.

[16] H. Riski and D. W. Utomo, "Algoritma Principal Component Analysis (PCA) dan Metode Bounding Box pada Pengenalan Citra Wajah," *J. Inform. J. Pengemb. IT*, vol. 9, no. 1, pp. 72–77, 2024, doi: 10.30591/jpit.v9i1.6165.

[17] J. Wang and S. Lee, "Data augmentation methods applying grayscale images for convolutional neural networks in machine vision," *Appl. Sci.*, vol. 11, no. 15, 2021, doi: 10.3390/app11156721.

[18] Y. S. Chernyshova, M. A. Aliev, E. S. Gushchanskaia, and A. V. Sheshkus, "Optical font recognition in smartphone-captured images and its applicability for ID forgery detection," vol. 1, p. 59, 2019, doi: 10.1117/12.2522955.

[19] G. Li, M. Zhang, J. Li, F. Lv, and G. Tong, "Efficient densely connected convolutional neural networks," *Pattern Recognit.*, vol. 109, 2021, doi: 10.1016/j.patcog.2020.107610.

[20] M. F. Gunardi, "Implementasi Augmentasi Citra pada Suatu Dataset," *J. Inform.*, vol. 9, no. 1, pp. 1–5, 2023.
[21] Tukino, "Penerapan Algoritma Convolutional Neural Network Untuk Klasifikasi Sentimen Pada Layanan e-Commerce," *J. DESAIN DAN Anal. Teknol.*, vol. 04, no. 1, pp. 44–53, 2025.
[22] M. Fadli and R. A. Saputra, "Klasifikasi Dan Evaluasi Performa Model Random Forest Untuk Prediksi Stroke," *JT J. Tek.*, vol. 12, no. 02, pp. 72–80, 2023, [Online]. Available: http://jurnal.umt.ac.id/index.php/jt/index
[23] M. Toyib, T. Decky, and K. Pratama, "Penerapan Algoritma CNN Untuk Mendeteksi Tulisan Tangan Angka Romawi dengan Augmentasi Data," *J. Mat. Ilmu Pengetah. Alam, Kebumian dan Angkasa*, vol. 2, no. 3, pp. 108–120, 2024, [Online]. Available: https://doi.org/10.62383/algoritma.v2i3.69
[24] Y. Rimal and N. Sharma, "Ensemble machine learning prediction accuracy: local vs. global precision and recall for multiclass grade performance of engineering students," *Front. Educ.*, vol. 10, no. April, pp. 1–16, 2025, doi: 10.3389/feduc.2025.1571133.