

APPLICATION OF CANNY OPERATOR IN BATIK MOTIF IMAGE CLASSIFICATION WITH CONVOLUTIONAL NEURAL NETWORK APPROACH

Iwan Jaya Bakti¹, Nirwana Hendrastuty^{*2}

^{1,2}Informatics, Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia, Indonesia
Email: ¹iwan_jaya_bakti@teknokrat.ac.id, ²nirwanahsndrastuty@teknokrat.ac.id

(Article received: February 22, 2024; Revision: March 20, 2024; published: May 28, 2024)

Abstract

Batik, as Indonesia's cultural heritage, has high artistic value and has a variety of unique motifs.. The main focus of this research is to solve the problem of the complexity and diversity of motifs found in Indonesian batik culture. The Canny operator is used as a first step to extract the edges of batik motifs, with the aim of improving the quality of feature extraction before entering the classification stage using CNN, specifically by using the DenseNet121 model. The dataset of this study was obtained through the Kaggle platform, published by Dionisius Darryl Hermansyah. The platform consists of 983 images (.jpg) with 20 different Indonesian batik motifs. Pre-processing includes the use of Canny for edge detection and data augmentation to increase the diversity of the dataset. Next, variations in the number of epochs and batch size were used to train the model. The results show that in the first test, the use of the Canny operation gives a higher confidence level in the model. In the model with Canny, there is a 1.6% increase in accuracy (33.57% with Canny and 31.97% without Canny). In addition, there are differences in the level of confidence in some batik classes. For example, the "batik mega mendung" class shows an increase in confidence of 66.57% with Canny (88.53% with Canny and 21.96% without Canny), while the "batik sekar" class shows a decrease in confidence of 12.09% with Canny.

Keywords: Batik Motif, Canny Operator, Classification, Convolutional Neural Network, DenseNet121.

PENERAPAN OPERATOR CANNY DALAM KLASIFIKASI CITRA MOTIF BATIK DENGAN PENDEKATAN CONVOLUTIONAL NEURAL NETWORK

Abstrak

Batik, sebagai warisan budaya Indonesia, memiliki nilai seni tinggi dan memiliki beragam motif yang unik. Fokus utama penelitian ini adalah untuk memecahkan masalah kompleksitas dan perbedaan motif yang ditemukan dalam budaya batik Indonesia. Operator Canny digunakan sebagai langkah awal untuk mengekstraksi tepi motif batik, dengan tujuan untuk meningkatkan kualitas ekstraksi fitur sebelum memasuki tahap klasifikasi menggunakan Convolutional Neural Network (CNN), khususnya dengan menggunakan model DenseNet121. Dataset penelitian ini diperoleh melalui platform Kaggle, yang diterbitkan oleh Dionisius Darryl Hermansyah. Platform ini terdiri dari 983 gambar (.jpg) dengan 20 motif batik Indonesia yang berbeda. Pre-processing mencakup penggunaan Canny untuk deteksi tepi dan augmentasi data untuk meningkatkan keragaman dataset. Selanjutnya, variasi jumlah epoch dan ukuran batch digunakan untuk melatih model. Hasil penelitian menunjukkan bahwa pada pengujian pertama, penggunaan operasi Canny memberikan tingkat confidence yang lebih tinggi pada model. Pada model dengan Canny, terdapat peningkatan akurasi sebesar 1.6% (33.57% dengan Canny dan 31.97% tanpa Canny). Selain itu, terlihat perbedaan tingkat confidence pada beberapa kelas batik. Sebagai contoh, kelas "batik mega mendung" menunjukkan peningkatan confidence sebesar 66.57% dengan Canny (88.53% dengan Canny dan 21.96% tanpa Canny), sedangkan kelas "batik sekar" menunjukkan penurunan confidence sebesar 12.09% dengan Canny (30.28% dengan Canny dan 17.19% tanpa Canny).

Kata kunci: DenseNet121, Convolutional Neural Network, Klasifikasi, Motif Batik, Operator Canny.

1. PENDAHULUAN

Batik, sebagai warisan budaya Indonesia, memiliki nilai seni tinggi dan memiliki beragam motif yang unik [1]. Pengenalan dan klasifikasi motif batik menjadi sebuah tantangan karena adanya variasi

bentuk dan warna. Oleh karena itu, penerapan teknologi pengolahan citra, khususnya menggunakan operator Canny dan Convolutional Neural Network (CNN), dapat memberikan kontribusi besar dalam upaya klasifikasi motif batik. Dalam konteks seni dan

budaya, batik menjadi bagian integral dari identitas Indonesia. Motif batik tidak hanya mencerminkan keindahan estetika, tetapi juga mengandung makna simbolik dan sejarah [2]. Keanekaragaman motif batik memberikan gambaran tentang kekayaan warisan budaya yang perlu dilestarikan dan dijaga. Meskipun telah ada upaya untuk mendokumentasikan dan menjaga keaslian motif batik, namun pengolahan data secara manual memiliki keterbatasan dalam hal kecepatan dan akurasi.

Dalam konteks pengolahan citra, CNN telah menjadi salah satu pendekatan utama dalam pengenalan pola dan klasifikasi. CNN terdiri dari lapisan-lapisan konvolusi yang secara efektif dapat mengekstrak fitur-fitur hierarkis dari data citra, memungkinkan model untuk memahami informasi yang semakin kompleks pada tingkat yang berbeda [3]. Kemampuan CNN untuk secara otomatis mempelajari representasi-fitur dari data *input* membuatnya sangat sesuai untuk tugas-tugas pengenalan pola seperti klasifikasi citra dengan motif yang bervariasi [4].

Sementara itu, operator Canny adalah metode deteksi tepi dalam pengolahan citra, telah terbukti efektif dalam mengekstrak fitur-fitur penting pada citra [5]. Operator ini bekerja dengan mengidentifikasi tempat-tempat di mana intensitas citra mengalami perubahan tajam, yang sering kali menandakan batas-batas objek dalam citra [6]. Langkah-langkah operator Canny, seperti pembulatan sudut arah tepi, penekanan non-maksimum, dan penggunaan *thresholding* untuk mengumpulkan piksel tepi, membantu menghasilkan tepi yang jelas dan akurat dalam citra. [7] Penggunaan operator Canny sebagai langkah awal dalam proses pengolahan citra, sebelum tahap klasifikasi menggunakan CNN, bertujuan untuk meningkatkan kualitas ekstraksi fitur dan memfasilitasi identifikasi pola yang kompleks dalam motif batik.

Penelitian terdahulu berfokus pada otomatisasi klasifikasi motif batik menggunakan *deep learning* dengan CNN. Penelitian ini menggunakan 598 citra batik berjenis ceplok, kawung, lereng, nitik, dan parang digunakan untuk melatih model. Peneliti menggunakan *Gray Level Co-occurrence Matrix* (GLCM) sebagai ekstraksi fitur. Hasilnya menunjukkan akurasi sebesar 66.14% untuk dataset baru lebih unggul dibandingkan dengan yang lama, dan augmentasi data tidak signifikan meningkatkan akurasi [8]. Tetapi masih terdapat celah penggunaan metode ekstraksi fitur lainnya dan memperbanyak kelas citra batik.

Beberapa penelitian terdahulu telah mendokumentasikan penggunaan operator Canny dalam berbagai konteks pengolahan citra diantaranya adalah penelitian tentang deteksi tepi pada *medical image* menggunakan Canny. Metode yang digunakan melibatkan pembulatan sudut arah tepi, penekanan non-maksimum, pelacakan tepi dengan histeresis, perhitungan probabilitas, dan pengukuran pelacakan

tepi. Hasil dibandingkan menggunakan nilai MSE dan PSNR, menunjukkan metode yang diusulkan mengungguli metode tradisional lain dalam mendeteksi tepi pada pemindaian CT paru-paru, sinar-X toraks, dan gambar mamografi [9].

Penelitian lain menyajikan implementasi algoritma deteksi tepi Canny untuk gambar ber-*noise* dengan menggunakan filter median untuk mengurangi *noise* dan meningkatkan akurasi. Algoritma Canny melibatkan langkah-langkah seperti deteksi noise, menghitung gradien gambar, penekanan non-maksima, penggandaan ambang, dan pelacakan tepi dengan histeresis. Hasil menunjukkan bahwa metode yang diusulkan ini dapat efektif mengatasi gangguan noise, mempertahankan data tepi, dan meningkatkan ketepatan deteksi tepi [10].

Kemudian Penelitian lain membahas penggunaan CNN dan Canny untuk mengklasifikasikan simbol-simbol stenografi tulisan tangan. Model CNN mencapai akurasi 95% dalam pelatihan dan validasi, dengan akurasi rata-rata 77% menggunakan Koefisien Kappa pada data uji. Hasil menunjukkan potensi penggunaan teknologi modern untuk mengenali dan mengklasifikasikan tulisan stenografi historis, dengan aplikasi potensial dalam sidang pengadilan dan jurnalisme. Canny meningkatkan akurasi klasifikasi, menunjukkan bahwa penggunaan algoritma ini dapat membantu dalam proses pengenalan simbol-simbol stenografi [11].

Selanjutnya, penelitian untuk mengidentifikasi 6 jenis buah durian Filipina dengan menggunakan CNN dan Canny. algoritma Deteksi Tepi Canny digunakan untuk mendeteksi tepi buah Durian pada gambar yang diambil. Tahapan yang digunakan adalah menghitung magnitudo dan sudut gradien tepi, kemudian menerapkan supresi non-maksimum untuk menemukan puncak gradien tepi. Selanjutnya, dilakukan *thresholding* untuk mengumpulkan piksel tepi yang akan dipertimbangkan dalam rentang nilai ambang yang ditentukan. Piksel tepi yang lemah ditolak sementara piksel tepi yang kuat dipertahankan. Eksperimen dilakukan dengan mengambil gambar buah Durian di tempat umum, dan hasilnya menunjukkan tingkat akurasi sebesar 71,429% [12].

Penelitian lain mendeteksi goresan pada gambar menggunakan metode deteksi tepi dan identifikasi goresan. Metode yang diusulkan menggunakan jaringan saraf konvolusi dan berhasil mencapai akurasi tinggi sebesar 97.1%. Selain itu, penelitian ini juga mengusulkan pendekatan deteksi tepi Canny berbasis CNN untuk mendeteksi goresan pada mobil menggunakan pemrosesan gambar dan *deep learning*. Sistem yang diusulkan mencapai akurasi deteksi tinggi sebesar 97.31% dan mengurangi *noise* dalam gambar sebesar 12.76% [13].

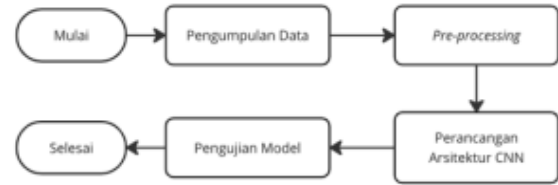
Selain itu terdapat juga penelitian tentang penggunaan CNN untuk mengklasifikasikan orang dengan *normal cognition* (CN), *early mild cognitive*

impairment (EMCI), dan *late mild cognitive impairment* (LMCI) berdasarkan data MRI. Sebelum menggunakan arsitektur CNN dengan tiga lapisan konvolusi, langkah-langkah *preprocessing* seperti normalisasi, pengambilan sampel, dan penghalusan gambar materi abu-abu dilakukan. Melalui klasifikasi yang akurat dari berbagai tahap gangguan kognitif, penelitian ini bertujuan untuk membantu diagnosis dini penyakit Alzheimer. Hasil penelitian menunjukkan bahwa kelompok CN dan LMCI memiliki klasifikasi CNN terbaik secara keseluruhan pada tampilan sagital dengan akurasi 94,54%. Pasangan EMCI/LMCI memiliki akurasi 93,96% dan CN/EMCI memiliki akurasi 93,30% [14].

Penggunaan operator Canny sebagai langkah awal dalam proses klasifikasi dapat meningkatkan kualitas ekstraksi fitur, sehingga memudahkan CNN untuk mengidentifikasi pola yang kompleks pada motif batik. CNN telah menjadi andalan dalam pengolahan citra dan pengenalan pola. Kekuatan CNN terletak pada kemampuannya untuk secara otomatis mempelajari fitur-fitur hierarkis dari data *input*, yang membuatnya sangat sesuai untuk tugas klasifikasi bermotif yang kompleks [15]. Penelitian ini mencoba mengisi celah pengetahuan dengan fokus pada penerapan operator Canny sebagai langkah awal dalam pengolahan citra motif batik sebelum masuk ke tahap klasifikasi menggunakan CNN. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan teknologi untuk melestarikan dan mengenali motif batik secara efisien.

2. METODE PENELITIAN

Penelitian ini dilakukan dengan mengikuti kerangka kerja untuk memastikan bahwa penelitian ini terstruktur dan terarah. Tahapan kerangka kerja penelitian dimulai dari studi literatur, pengumpulan data citra batik dan pengembangan sistem yang mampu mengklasifikasikan citra batik seperti pada Gambar 1.



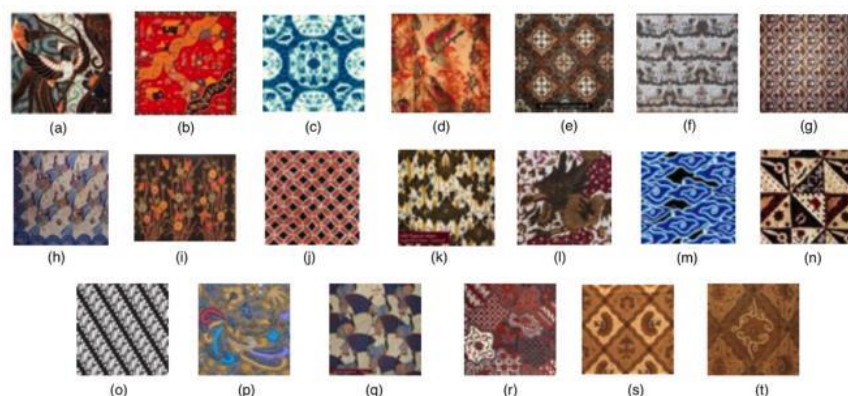
Gambar 1. Kerangka Penelitian

2.1. Studi Literatur

Studi literatur menjadi langkah awal yang kritis dalam penelitian ini. Pemahaman mendalam terkait pengolahan citra, metode deteksi tepi menggunakan operator Canny, konsep augmentasi data, dan prinsip dasar CNN akan membantu mengidentifikasi gap pengetahuan dan memastikan keberlanjutan penelitian ini dalam kerangka ilmiah yang kokoh. Analisis literatur juga akan membantu mengidentifikasi teknik-teknik terkini dalam pengenalan motif pada citra dan memberikan dasar yang kuat untuk merancang metodologi penelitian yang tepat.

2.2. Pengumpulan Data

Dataset yang digunakan dalam penelitian ini diperoleh melalui platform Kaggle, sebuah sumber data terbuka yang menyediakan berbagai dataset untuk keperluan penelitian dan pengembangan. Pemilihan dataset dari Kaggle dilakukan dengan pertimbangan untuk mendapatkan variasi motif batik yang luas, mencakup berbagai daerah di Indonesia. Dataset yang diterbitkan oleh Dionisius Darryl Hermansyah di Kaggle terdiri dari 983 citra (.jpg) dengan 20 motif batik Indonesia yang beragam. Data ini dikumpulkan melalui proses *image-scraping* menggunakan layanan Bing.com oleh Ultralytics. Proses pemecahan dataset dilakukan dengan membagi dataset menjadi dua bagian utama: data latih (*training data*) dan data uji (*testing data*). Adapun pembagian data ini dilakukan dengan proporsi 80% untuk data latih dan 20% untuk data uji. Contoh dari data batik yang digunakan dapat dilihat pada Gambar 2.



Gambar 2. Sampel Citra Batik (a) Bali, (b) Betawi, (c) Celup, (d) Cendrawasih, (e) Ceplok, (f) Ciamis, (g) Sogan, (h) Garutan, (i) Gentongan, (j) Kawung, (k) Keraton, (l) Lasem, (m) Megamendung, (n) Tambal, (o) Parang, (p) Pekalongan, (q) Priangan, (r) Sekar, (s) Sidoluhur, (t) Sidomukti

2.3. Pre-processing

Tahap *pre-processing* data melibatkan komponen utama yaitu penerapan operator Canny untuk deteksi tepi dan augmentasi data untuk meningkatkan keragaman dataset. Operator Canny dipilih dengan untuk membantu dalam mengidentifikasi tepian objek citra dengan tingkat akurasi yang tinggi, memberikan informasi visual yang lebih jelas dan relevan kepada model CNN [16]. Selain itu, untuk meningkatkan kemampuan generalisasi model, teknik augmentasi data seperti *width_shift_range*, *height_shift_range*, dan *zoom_range* digunakan. Dengan mengimplementasikan augmentasi data, variasi kecil pada posisi dan skala citra dapat dipertimbangkan selama proses pelatihan dan menghasilkan dataset yang lebih beragam. Hal ini bukan hanya membantu meningkatkan kinerja model pada data yang belum pernah dilihat sebelumnya, tetapi juga berperan dalam mencegah *overfitting*, sehingga model lebih adaptif terhadap variasi yang mungkin terjadi pada citra batik yang sebenarnya [17]. Dengan demikian, tahap *pre-processing* yang holistik dan cermat ini memastikan bahwa model CNN dapat memanfaatkan informasi tepi yang akurat dan dihadapkan pada berbagai situasi dengan keberagaman yang cukup, menjadikannya lebih *robust* dan responsif terhadap variasi motif batik yang ada.

2.4. Perancangan Arsitektur CNN

Perancangan arsitektur dalam penelitian ini mengusulkan penggunaan model DenseNet. DenseNet bertujuan untuk meningkatkan kinerja tugas target dengan menerapkan *transfer learning*, yaitu mentransfer pengetahuan yang telah dipelajari dari tugas sumber ke tugas target [18]. DenseNet merupakan jenis arsitektur CNN yang memperkenalkan konsep konektivitas yang padat antara lapisan-lapisan. Dalam DenseNet, setiap lapisan menerima input dari semua lapisan sebelumnya dalam satu blok atau satuan. Hal ini menghasilkan aliran informasi yang lebih langsung dan memungkinkan jaringan untuk memanfaatkan kembali fitur-fitur yang telah dipelajari lebih efisien.

2.5. Pengujian Model

Pengujian model ini dilakukan dengan menggunakan gambar batik yang diperoleh dari hasil pencarian di Google yang tidak ada dalam dataset. Proses pengujian ini bertujuan untuk menguji sejauh mana kemampuan model CNN dalam mengklasifikasikan citra batik yang tidak terlibat dalam proses pelatihan. Keempat gambar tersebut dipilih dengan tujuan untuk menciptakan variasi yang lebih luas dan menantang bagi model.

Pada setiap gambar, model CNN akan melakukan prediksi terkait dengan motif batik yang terdapat pada gambar tersebut. Hasil prediksi tersebut kemudian dievaluasi untuk menilai keakuratan dan

ketepatan model dalam mengidentifikasi motif batik. *Confusion Matrix* adalah metode yang umum digunakan untuk mengevaluasi pengukuran kinerja klasifikasi. *Confusion matrix* menggambarkan hasil klasifikasi berdasarkan empat kemungkinan hasil: *True Positive* (TP) - jumlah data positif yang diklasifikasikan dengan benar, *True Negative* (TN) - jumlah data negatif yang diklasifikasikan dengan benar, *False Positive* (FP) - jumlah data negatif yang diklasifikasikan secara tidak benar sebagai positif, dan *False Negative* (FN) - jumlah data positif yang diklasifikasikan secara tidak benar sebagai negatif. Akurasi, presisi, recall, dan skor F1 digunakan sebagai metrik untuk mengevaluasi model. Persamaan untuk menghitung akurasi, presisi, recall, dan skor F1 diwakili oleh Persamaan 1 hingga Persamaan 4.

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

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

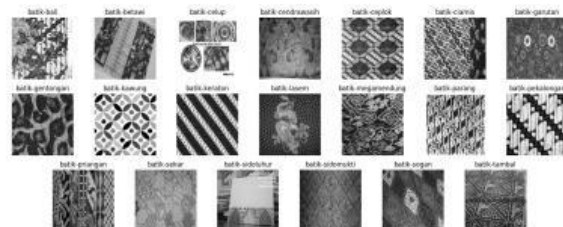
$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1\ Score = 2 \frac{(recall*precision)}{(recall+precision)} \quad (4)$$

3. HASIL DAN PEMBAHASAN

3.1. Persiapan Data Awal

Tahapan pertama yang dilakukan adalah membuat citra dalam mode *grayscale* dengan menggunakan. Ini penting karena mode *grayscale* memungkinkan representasi citra dengan satu saluran warna saja, yaitu tingkat keabuan. Dalam konteks ini, warna tidak diperlukan untuk analisis, dan mode *grayscale* memungkinkan representasi yang lebih sederhana dan efisien. Setelah citra dibaca dalam mode *grayscale*, langkah selanjutnya adalah mengubah ukurannya menjadi 224 x 224.

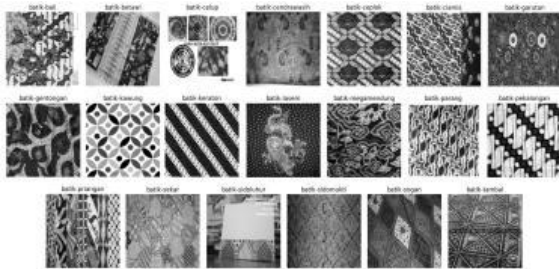


Gambar 3. Citra Setelah *Resize* dan *Grayscale*

3.2. Hasil Penerapan Canny

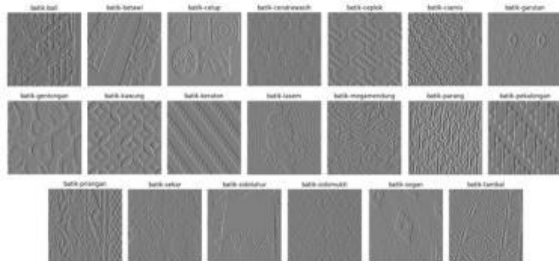
Penerapan operator Canny bertujuan untuk mengekstraksi tepi pada citra dengan memanfaatkan teknik pengurangan *noise*, perhitungan gradien, pemilihan tepi terlokalisasi, dan penerapan histeresis. Tahap pertama citra input melewati proses

penghalusan (*smoothing*) menggunakan filter *Gaussian*. Proses ini bertujuan untuk mengurangi noise yang mungkin ada dalam citra dan secara bersamaan mempertahankan tepi yang signifikan. Filter Gaussian bekerja dengan menerapkan kernel Gaussian pada setiap piksel dalam citra, di mana nilai piksel baru dihitung sebagai rata-rata tertimbang dari nilai piksel di sekitarnya, dengan bobot tertinggi diberikan pada piksel yang lebih dekat dengan piksel yang sedang diproses. Hasil dari proses ini adalah citra yang lebih halus, di mana noise telah dikurangi, tetapi tepi yang signifikan tetap dipertahankan dengan baik [19], dengan hasil yang bisa dilihat pada Gambar 4.

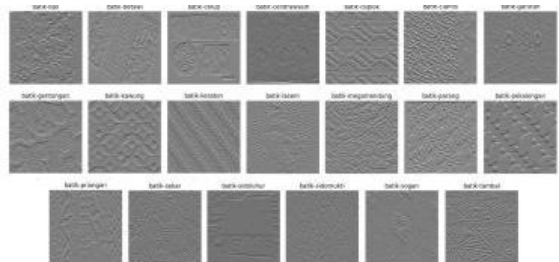


Gambar 4. Hasil Penerapan Gaussian Filter

Setelah proses penghalusan menggunakan filter Gaussian, tahap selanjutnya adalah melakukan perhitungan gradien untuk menentukan di mana tepi yang signifikan mungkin terletak dalam citra. Proses perhitungan gradien ini melibatkan konvolusi citra dengan kernel Sobel secara terpisah untuk mendapatkan gradien vertikal (Y) dan gradien horizontal (X) seperti pada Gambar 5 dan Gambar 6. Gradien vertikal mengukur perubahan intensitas piksel sepanjang sumbu vertikal, sementara gradien horizontal mengukur perubahan intensitas sepanjang sumbu horizontal.



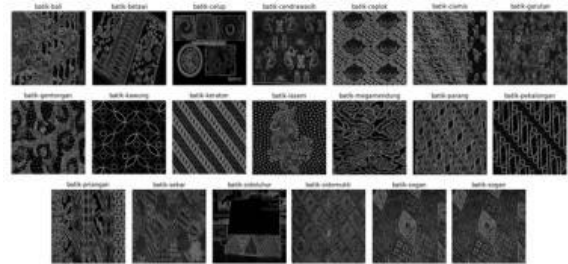
Gambar 5. Gradient X Yang Dihasilkan



Gambar 6. Gradient Y Yang Dihasilkan

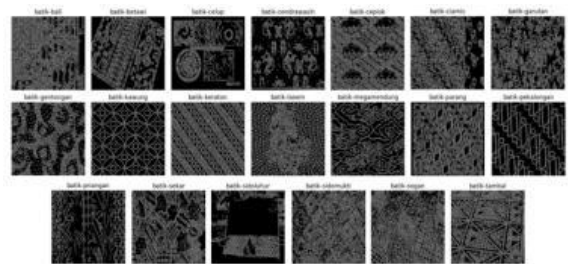
Setelah kedua gradien ini dihitung, mereka digabungkan untuk menghitung magnitudo gradien,

yang menunjukkan seberapa signifikan perubahan intensitas piksel dalam citra tersebut. Hasil dari tahap ini adalah citra gradien, di mana intensitas piksel merepresentasikan seberapa besar perubahan intensitas terjadi pada titik tersebut. Pada citra gradien, daerah-daerah dengan perubahan intensitas yang signifikan akan memiliki intensitas yang lebih tinggi, dengan hasil seperti pada Gambar 7.



Gambar 7. Magnitudo Gradient Yang Dihasilkan

Tepi yang dihasilkan dari perhitungan gradien kemudian disaring untuk mempertahankan hanya tepi yang terlokalisasi. Proses ini dilakukan dengan menggunakan metode non-maximum suppression. Teknik non-maximum suppression bekerja dengan membandingkan magnitudo gradien di setiap titik tepi dengan magnitudo gradien di sekitarnya dalam arah normal terhadap garis gradien. Proses ini menghasilkan citra dengan tepi yang lebih tajam dan terlokalisasi, di mana hanya tepi yang signifikan dipertahankan, sementara tepi yang lemah atau terkontaminasi oleh noise dihilangkan. Hasil dari tahap ini akan direpresentasikan dalam Gambar 8.

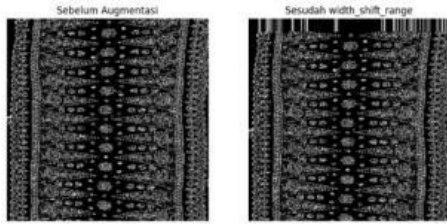


Gambar 8. Hasil Ekstraksi Tepi

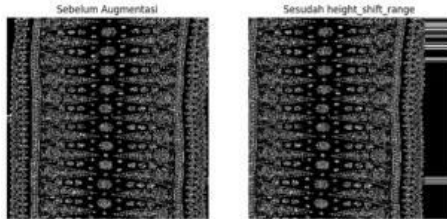
3.3. Hasil Augmentasi

Dalam penelitian ini, augmentasi dilakukan menggunakan *width_shift_range* dan *height_shift_range* dengan nilai diatur ke 0.2 atau sama dengan 20%. Dalam hal ini, gambar digeser hingga 20% dari lebar dan tinggi gambar. Pergeseran ini berguna untuk mensimulasikan variasi posisi motif batik dalam citra seperti yang bisa dilihat pada Gambar 9 dan Gambar 10.

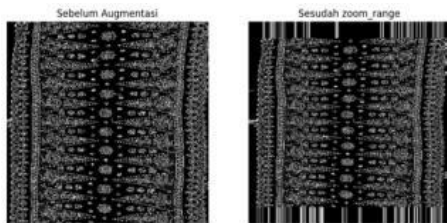
Augmentasi juga dilakukan menggunakan *zoom_range*, nilai diatur sebesar 20%. Dengan nilai ini, gambar dapat di *zoom* hingga 20%. Ini dapat membantu model mengatasi variasi dalam jarak pemotretan atau perbedaan resolusi citra dengan hasil seperti pada Gambar 11.



Gambar 9. Sampel Hasil Augmentasi Width Shift Range



Gambar 10. Sampel Hasil Augmentasi Height Shift Range



Gambar 11. Hasil Augmentasi Zoom Range

3.4. Perancangan Arsitektur

Dalam hal ini, model DenseNet akan digunakan karena kemampuannya mengatasi masalah *vanish gradient* dalam arsitektur CNN, dan telah berhasil diimplementasikan dalam berbagai penelitian sebelumnya [20], dengan konfigurasi seperti pada Tabel 2.

Tabel 2. Konfigurasi DenseNet

Layer	Config
Input Layer	Input shape: (224, 224, 3)
DenseNet121 Base Model	Weight initialization: 'imagenet' Include top: False
GlobalAveragePooling2D Layer	
Dense Layer (Hidden)	Number of units: 512 Activation function: ReLU
Dropout Layer	Dropout rate: 0.2
Dense Layer (output)	Number of units: 20 Activation function: Softmax

Model pada Tabel 2 yang diimplementasikan menggunakan arsitektur DenseNet121 memiliki konfigurasi yang terdiri dari lapisan-lapisan. Dimulai dengan lapisan *input* yang menentukan dimensi gambar (224x224 piksel dengan tiga saluran warna RGB), model menggunakan DenseNet121 sebagai *base model* untuk *transfer learning* dengan inialisasi bobot dari dataset ImageNet dan mengabaikan *fully-connected* layer teratas [21]. *Global Average Pooling2D* digunakan untuk menggantikan Flatten, diikuti oleh lapisan Dense dengan 512 unit dan aktivasi ReLU untuk ekstraksi fitur yang lebih kompleks [22]. Penambahan *Dropout* dengan tingkat 0.2 membantu mencegah *overfitting* dengan secara acak menonaktifkan sebagian unit

selama pelatihan [23]. Lapisan *output* terakhir memiliki 20 unit (sesuai dengan jumlah kelas aksara Lampung) dan fungsi aktivasi *softmax*, menghasilkan distribusi probabilitas untuk setiap kelas.

Total parameter dalam model ini mencapai 7,572,564, dengan 535,060 parameter yang dapat dilatih, termasuk parameter dari lapisan Dense yang dapat diubah selama pelatihan, dan 7,037,504 parameter yang tidak dapat dilatih dari base model DenseNet121 yang diinisialisasi dengan bobot dari dataset ImageNet, dengan ringkasan yang dapat dilihat pada Gambar 12.

```

Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
densenet121 (Functional)    (None, 7, 7, 1024)       7037504
global_average_pooling2d ( (None, 1024)             0
GlobalAveragePooling2D)
dense (Dense)                (None, 512)              524800
dropout (Dropout)           (None, 512)              0
dense_1 (Dense)              (None, 20)               10260
-----
Total params: 7572564 (28.89 MB)
Trainable params: 535060 (2.04 MB)
Non-trainable params: 7037504 (26.85 MB)
    
```

Gambar 12. Ringkasan Model.

3.5. Training Model

Training Model dilakukan menggunakan model dari DenseNet. Dalam penelitian ini, dilakukan tuning parameter, dimana fungsi *tuning* ini untuk mendapatkan hasil akurasi terbaik dari kombinasi parameter *epoch* dan *batch* dari model [24]. Training dilakukan terhadap dua jenis model. Model pertama ditraining dengan citra yang diekstraksi fitur tepinya menggunakan *canny*, kemudian model kedua ditraining menggunakan data asli tanpa ekstraksi tepi. Hasil dari kedua training dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 3. Hasil Training Data Canny

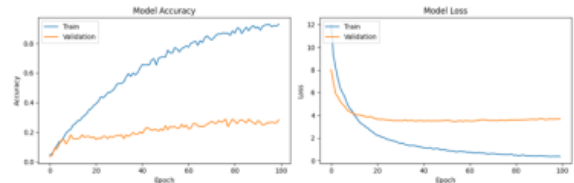
No	Epoch	Batch Size	Val Accuracy
1	50	16	0.3096
2	50	32	0.3197
3	50	64	0.3299
4	100	16	0.3357
5	100	32	0.3350
6	100	64	0.3045

Tabel 4. Hasil Training Data Tanpa Canny

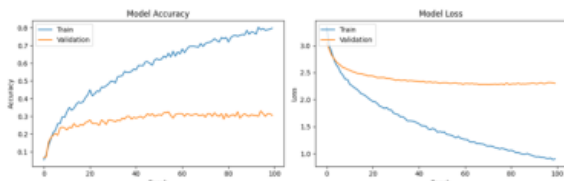
No	Epoch	Batch Size	Val Accuracy
1	50	16	0.2690
2	50	32	0.3147
3	50	64	0.2538
4	100	16	0.3197
5	100	32	0.2893
6	100	64	0.2842

Pada Tabel 3, peningkatan jumlah *epoch* meningkatkan kinerja model. Ini dapat terlihat dari perbandingan antara hasil *epoch* 50 dan 100 pada *batch size* 16 dan 32, di mana *val accuracy* meningkat atau tetap relatif stabil. *Batch size* 64 memiliki *val accuracy* yang lebih rendah dibandingkan dengan *batch size* 16 dan 32 pada *epoch* 50. Namun, pada

epoch 100, *batch size* 64 menghasilkan *val accuracy* yang lebih tinggi dibandingkan dengan *batch size* 16 dan 32. Terdapat variasi dalam kinerja model ketika menggunakan metode Canny. Pada beberapa kasus, menggunakan Canny (seperti pada *epoch* 50 dengan *batch size* 32) menghasilkan peningkatan *val accuracy*, sedangkan pada Tabel 4 mencerminkan performa model tanpa Canny, menunjukkan bahwa *val accuracy* cenderung lebih rendah secara umum. Grafik hasil model terbaik dari kedua eksperimen pelatihan dapat dilihat pada Gambar 13 dan Gambar 14.



Gambar 14. Grafik Pelatihan Model Tanpa Canny



Gambar 13. Grafik Pelatihan Model Canny

3.6. Pengujian Model

Pada proses pelatihan sebelumnya menghasilkan sebuah model, yang kemudian dapat digunakan untuk tujuan klasifikasi. Pada penelitian ini, model terbaik dari eksperimen menghasilkan dua model, yaitu model dengan penerapan canny dan model tanpa canny. Model tersebut kemudian diuji dengan menggunakan gambar batik yang diperoleh dari hasil pencarian di Google yang tidak ada dalam dataset. Hasil dari pengujian model pertama adalah dengan penerapan canny dapat dilihat pada Gambar 15.

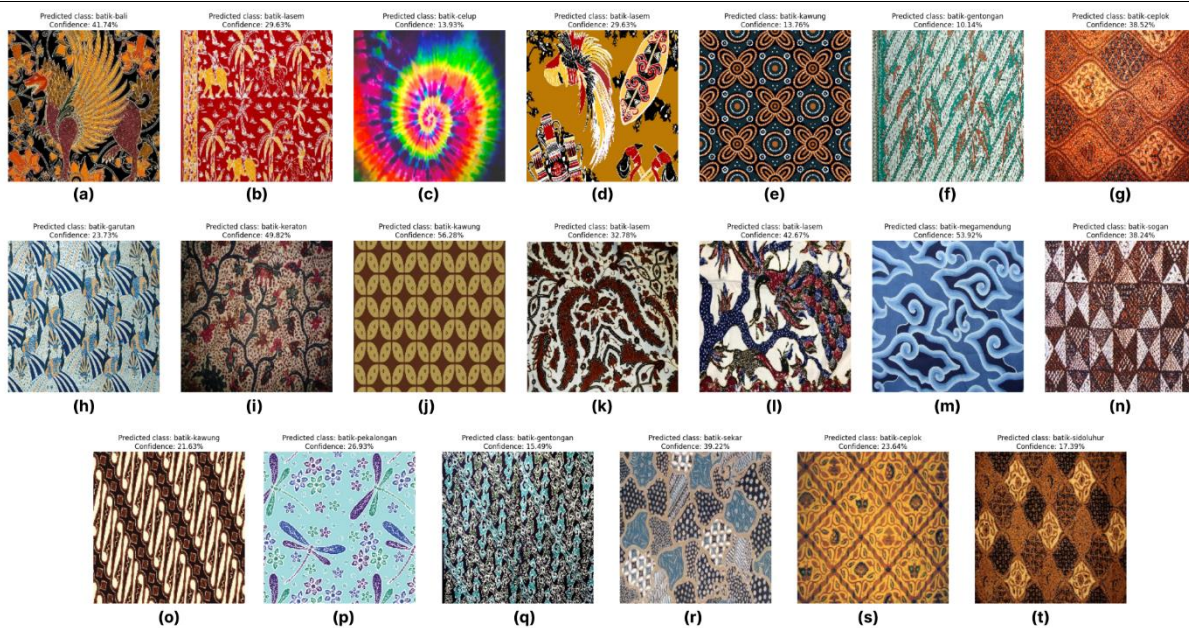


Gambar 15. Hasil Pengujian Model Dengan Canny (a) Bali, (b) Betawi, (c) Celup, (d) Cendrawasih, (e) Ceplok, (f) Ciamis, (g) Sogan, (h) Garutan, (i) Gentongan, (j) Kawung, (k) Keraton, (l) Lasem, (m) Megamendung, (n) Tambal, (o) Parang, (p) Pekalongan, (q) Priangan, (r) Sekar, (s) Sidoluhur, (t) Sidomukti

Berdasarkan Gambar 15, hasil pengujian model klasifikasi citra batik dengan menerapkan metode Canny Edge Detection menunjukkan performa yang cukup baik. Dari total 20 pengujian, sebanyak 16 prediksi berhasil diklasifikasikan dengan benar, sementara 4 prediksi lainnya salah. Meskipun demikian, terdapat beberapa prediksi yang memiliki tingkat kepercayaan (confidence) yang rendah, seperti pada contoh "Tambal" dan "Parang", di mana confidence-nya kurang dari 15%. Hal ini menunjukkan bahwa model cenderung tidak yakin dalam melakukan prediksi pada kasus-kasus tertentu.

Meskipun demikian, terdapat juga prediksi dengan tingkat kepercayaan yang cukup tinggi, seperti "Megamendung" dengan 88.71% dan "Celup" dengan 78.29%, yang ternyata benar. Namun, terdapat kasus di mana prediksi salah meskipun memiliki confidence yang cukup tinggi, seperti contoh "Ceplok" dengan 34.61% yang diprediksi sebagai "Ceplok", namun seharusnya merupakan "Keraton".

Pengujian ke dua dilakukan menggunakan model tanpa penerapan canny yang dapat dilihat pada Gambar 16.



Gambar 15. Hasil Pengujian Model Tanpa Canny (a) Bali, (b) Betawi, (c) Celup, (d) Cendrawasih, (e) Ceplok, (f) Ciamis, (g) Sogan, (h) Garutan, (i) Gentongan, (j) Kawung, (k) Keraton, (l) Lasem, (m) Megamendung, (n) Tambal, (o) Parang, (p) Pekalongan, (q) Priangan, (r) Sekar, (s) Sidoluhur, (t) Sidomukti

Dalam pengujian model klasifikasi citra batik tanpa menerapkan metode Canny Edge Detection, hasilnya menunjukkan performa yang berbeda dengan pengujian sebelumnya. Dari total 20 pengujian, hanya 9 prediksi yang berhasil diklasifikasikan dengan benar, sementara 11 prediksi lainnya salah. Tingkat kepercayaan (confidence) pada prediksi-prediksi ini juga bervariasi, namun secara umum, terdapat kecenderungan tingkat kepercayaan yang lebih rendah dibandingkan dengan hasil pengujian sebelumnya. Beberapa prediksi yang salah juga menunjukkan adanya ketidakcocokan yang signifikan antara kelas prediksi dan kelas aktual citra batiknya. Misalnya, citra batik dengan kelas "Cendrawasih" diprediksi sebagai "Lasem" dengan confidence 29.63%, yang merupakan salah klasifikasi yang cukup signifikan. Hal ini menunjukkan bahwa model tanpa penerapan Canny Edge Detection mungkin memiliki kinerja yang lebih rendah dalam mengenali pola dan fitur pada citra batik..

4. DISKUSI

Penelitian ini berhasil menghasilkan model klasifikasi batik yang mampu mengidentifikasi 20 jenis batik dengan penerapan Canny sebagai ekstraksi fitur menggunakan CNN. Pelatihan dilakukan pada dua kondisi, yaitu dengan dan tanpa penerapan Canny. Selama pelatihan, dilakukan *tuning* parameter untuk mencari kombinasi terbaik dari *epoch* dan *batch size* pada masing-masing model.

Hasil penelitian menunjukkan bahwa pada model dengan penerapan Canny, parameter terbaik yang ditemukan adalah *epoch* 100 dan *batch size* 16, dengan akurasi mencapai 33.57%. Sedangkan pada model tanpa penerapan Canny, parameter terbaik

adalah *epoch* 100 dan *batch size* 16, dengan akurasi 31,97%.

Pada tahap pengujian dengan penerapan Canny, model menunjukkan tingkat *confidence* yang bervariasi untuk berbagai kelas batik. Sebagai contoh, beberapa prediksi memiliki tingkat kepercayaan yang tinggi, seperti pada contoh batik celup dengan *confidence* 78.29% dan batik megamendung dengan *confidence* 88.71%.

Pada pengujian model tanpa penerapan Canny, hasil eksperimen menunjukkan tingkat *confidence* yang lebih rendah untuk beberapa kelas batik tertentu. Dari 20 pengujian, hanya 9 prediksi yang benar dan 11 prediksi salah. Ini menunjukkan bahwa model memiliki tingkat akurasi sekitar 45%, yang jauh lebih rendah dibandingkan dengan pengujian sebelumnya yang menggunakan Canny.

Penelitian sebelumnya terkait klasifikasi citra batik menggunakan metode GLCM dan JST [25]. Terdapat beberapa kekurangan yang dapat diidentifikasi. Pertama, penelitian sebelumnya memiliki batasan dalam menghadapi kompleksitas motif batik yang bervariasi. Algoritma *backpropagation* digunakan untuk mengklasifikasi citra batik berdasarkan motif geometri dan non geometri dengan akurasi sebesar 85.71%. Penelitian tersebut tidak secara khusus membahas atau menerapkan teknik deteksi tepi seperti operator Canny, yang dapat memberikan wawasan tambahan terhadap detail tepi dan membantu memperbaiki tingkat *confidence*.

Selain itu, penelitian lain bertujuan untuk mengklasifikasi dan mengenali motif batik menggunakan CNN berdasarkan K-means Clustering [26]. Data citra batik Yogyakarta digunakan, terdiri dari tiga jenis batik: Batik Kawung Lama, Batik

Parang Barong, dan Batik Semen Sidomukti. Hasil penelitian menunjukkan bahwa penggunaan median filter memberikan hasil yang lebih baik daripada *sharpening*. Proses klasifikasi citra batik dengan CNN berdasarkan K-means *Clustering* dari hasil median filter memiliki akurasi 100%, sementara dari hasil *sharpening* memiliki akurasi 80%. Penelitian sebelumnya juga memiliki batasan dalam menghadapi kompleksitas motif batik yang bervariasi dibandingkan penelitian yang telah dilakukan.

5. KESIMPULAN

Berdasarkan hasil penelitian dapat disimpulkan bahwa penggunaan operasi Canny pada pengujian pertama memberikan hasil dengan tingkat *confidence* yang lebih tinggi. Operasi Canny tampaknya membantu model dalam mengekstrak fitur-fitur yang lebih diskriminatif, meningkatkan kemampuan model untuk mengklasifikasikan kelas-kelas tersebut. Di sisi lain, pada pengujian tanpa operasi Canny, model memberikan tingkat *confidence* yang lebih rendah, menunjukkan pengaruh signifikan dari proses pra-pemrosesan terhadap hasil prediksi. Perbedaan tingkat *confidence* antara kelas batik menyoroti kompleksitas dalam pengklasifikasian beberapa motif dan pola, dengan beberapa kelas mendapatkan prediksi yang lebih pasti dibandingkan dengan yang lain. Hasil eksperimen menunjukkan bahwa model dengan Canny mampu mengidentifikasi 20 jenis batik dengan akurasi sebesar 33.57%, sedangkan model tanpa Canny mencapai akurasi 31.97% dengan selisih peningkatan akurasi sebesar 1.6%.

Dengan menerapkan Canny, model klasifikasi citra batik mampu memberikan prediksi dengan tingkat kepercayaan yang lebih tinggi dan jumlah prediksi yang benar lebih dominan. Hal ini menunjukkan bahwa metode Canny memberikan kontribusi positif terhadap kemampuan model dalam mengenali pola dan fitur pada citra batik. Sebaliknya, tanpa penerapan Canny, performa model menurun dengan jumlah prediksi yang benar lebih rendah dan tingkat kepercayaan yang juga cenderung lebih rendah. Dengan demikian, kesimpulan yang dapat diambil adalah penerapan Canny merupakan langkah yang efektif dalam meningkatkan akurasi klasifikasi citra batik.

DAFTAR PUSTAKA

- [1] S. Saifulloh, H. A. Mumtahana, and M. W. Astuti, "Pelatihan Design Motif Batik Berbasis Kearifan Lokal Di Kelompok Batik Bogo Desa Kediren Kecamatan Lembeyan Kabupaten Magetan," *Jurnal Altifani Penelitian dan Pengabdian kepada Masyarakat*, vol. 2, no. 1, pp. 13–21, Jan. 2022, doi: 10.25008/altifani.v2i1.193.
- [2] G. P. Kurniawan, H. Almash, and A. Shofwana, "Analisis Makna Filosofis Motif Batik Ponorogo Sebagai Upaya Penanaman Pendidikan Karakter," *Attractive : Innovative Education Journal*, vol. 3, no. 2, 2021, [Online]. Available: <https://www.attractivejournal.com/index.php/aj/>
- [3] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0276-2.
- [4] M. Shafaey, M. ElBery, M. Salem, H. Moushier, E.-S. A. El-Dahshan, and M. Tolba, "HYPER SPECTRAL IMAGE ANALYSIS USING A CUSTOM SPECTRAL CONVOLUTIONAL NEURAL NETWORK," *International Journal of Intelligent Computing and Information Sciences*, vol. 22, no. 4, pp. 146–158, Dec. 2022, doi: 10.21608/ijicis.2022.147175.1198.
- [5] Y. Liu *et al.*, "Richer Convolutional Features for Edge Detection," *IEEE Trans Pattern Anal Mach Intell*, vol. 41, no. 8, pp. 1939–1946, Aug. 2019, doi: 10.1109/TPAMI.2018.2878849.
- [6] S. T. H. Kieu, A. Bade, and M. H. A. Hijazi, "Modified canny edge detection technique for identifying endpoints," in *Journal of Physics: Conference Series*, Institute of Physics, 2022. doi: 10.1088/1742-6596/2314/1/012023.
- [7] M. He, R. Zhang, S. Liu, Y. Tan, and Y. Zeng, "Ultrasonic Image Diagnosis of Liver and Spleen Injury Based on a Double-Channel Convolutional Neural Network," *Wirel Commun Mob Comput*, vol. 2021, 2021, doi: 10.1155/2021/2827011.
- [8] D. Gede, T. Meranggi, N. Yudistira, and Y. A. Sari, "INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION journal homepage : www.joiv.org/index.php/joiv INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION Batik Classification Using Convolutional Neural Network with Data Improvements." [Online]. Available: www.joiv.org/index.php/joiv
- [9] S. Widiyanto, D. Sundani, Y. Karyanti, and D. T. Wardani, "Edge Detection Based on Quantum Canny Enhancement for Medical Imaging," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, 2019. doi: 10.1088/1757-899X/536/1/012118.
- [10] E. A. Sekehravani, E. Babulak, and M. Masoodi, "Implementing canny edge detection algorithm for noisy image," *Bulletin of Electrical Engineering and*

- Informatics*, vol. 9, no. 4, pp. 1404–1410, Aug. 2020, doi: 10.11591/eei.v9i4.1837.
- [11] Amity University. Dubai Campus, Amity University. Dubai Campus. IEEE Student Branch, Institute of Electrical and Electronics Engineers. UAE Section, and Institute of Electrical and Electronics Engineers, *Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE' 2019) : 11--12 December 2019, venue: Amity University Dubai, UAE*.
- [12] J. N. Uy and J. F. Villaverde, “A Durian Variety Identifier Using Canny Edge and CNN,” in *2021 7th International Conference on Control Science and Systems Engineering, ICCSSE 2021*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021, pp. 293–297. doi: 10.1109/ICCSSE52761.2021.9545195.
- [13] A. Sasikumar, M. Sathyanarayanan, A. N. Sriyappan, R. Santhosh, and R. Reshma, “A CNN-based Canny Edge Detection Approach for Car Scratch Detection,” in *6th International Conference on Inventive Computation Technologies, ICICT 2023 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 404–409. doi: 10.1109/ICICT57646.2023.10134444.
- [14] H. T. Gorji and N. Kaabouch, “A deep learning approach for diagnosis of mild cognitive impairment based on mri images,” *Brain Sci*, vol. 9, no. 9, Sep. 2019, doi: 10.3390/brainsci9090217.
- [15] M. A. Rasyidi and T. Bariyah, “Batik pattern recognition using convolutional neural network,” *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1430–1437, Aug. 2020, doi: 10.11591/eei.v9i4.2385.
- [16] A. Baloch, T. D Memon, F. Memon, B. Lal, V. Viyas, and T. Jan, “Hardware Synthesize and Performance Analysis of Intelligent Transportation Using Canny Edge Detection Algorithm,” *International Journal of Engineering and Manufacturing*, vol. 11, no. 4, pp. 22–32, Aug. 2021, doi: 10.5815/ijem.2021.04.03.
- [17] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.
- [18] Y. D. Zhang, S. C. Satapathy, X. Zhang, and S. H. Wang, “COVID-19 Diagnosis via DenseNet and Optimization of Transfer Learning Setting,” *Cognit Comput*, 2021, doi: 10.1007/s12559-020-09776-8.
- [19] A. Imam Rauyani, M. Hamka Ibrahim, and S. Pramono, “ROI based Indonesian Paper Currency Recognition Using Canny Edge Detection.”
- [20] S. H. Noh, “Performance comparison of CNN models using gradient flow analysis,” *Informatics*, vol. 8, no. 3, Sep. 2021, doi: 10.3390/informatics8030053.
- [21] H. Alshazly, C. Linse, M. Abdalla, E. Barth, and T. Martinetz, “COVID-Nets: deep CNN architectures for detecting COVID-19 using chest CT scans,” *PeerJ Comput Sci*, vol. 7, pp. 1–40, 2021, doi: 10.7717/peerj-cs.655.
- [22] L. Lv, M. Peng, X. Wang, and Y. Wu, “Multi-scale information fusion network with label smoothing strategy for corneal ulcer classification in slit lamp images,” *Front Neurosci*, vol. 16, Nov. 2022, doi: 10.3389/fnins.2022.993234.
- [23] K. Sanjar, A. Rehman, A. Paul, and K. Jeonghong, “Weight dropout for preventing neural networks from overfitting,” in *2020 8th International Conference on Orange Technology, ICOT 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/ICOT51877.2020.9468799.
- [24] N. ALHAWAS and Z. TÜFEKÇİ, “The Effectiveness of Transfer Learning and Fine-Tuning Approach for Automated Mango Variety Classification,” *European Journal of Science and Technology*, Mar. 2022, doi: 10.31590/ejosat.1082217.
- [25] A. A. Kasim, J. Jurusan, I. Komputer, and D. Elektronika, “Klasifikasi Citra Batik Menggunakan Jaringan Syaraf Tiruan Berdasarkan Gray Level Co-Occurrence Matrices (GLCM) Agus Harjoko.”
- [26] A. P. A. Masa and H. Hamdani, “Klasifikasi Motif Citra Batik Menggunakan Convolutional Neural Network Berdasarkan K-means Clustering,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 4, p. 1292, Oct. 2021, doi: 10.30865/mib.v5i4.3246.