

KLASIFIKASI WARNA KAIN TENUN LOTIS MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK

Elike Adielwin Nenometa^{*1}, Skolastika Siba Igon²

^{1,2}Teknik Informatika Strata Satu, STIKOM Uyelindo Kupang, Indonesia

^{1*}elikenenometa@gmail.com, ²igon5kolastika@gmail.com

Abstrak

Kain tenun Lotis merupakan warisan budaya khas dari Kabupaten Timor Tengah Selatan, Nusa Tenggara Timur, yang berasal dari tiga suku utama: Amanuban, Amanatun, dan Mollo. Kain ini memiliki ciri khas pada motif geometris, corak warna alami, serta makna sosial dan budaya yang mendalam. Kain tenun ini umumnya digunakan dalam upacara adat, simbol status sosial, dan sebagai identitas etnis masyarakat setempat. Namun, pelestariannya menghadapi tantangan di era digital, terutama dalam proses klasifikasi dan dokumentasi karena kompleksitas visual yang dimiliki. Variasi warna dan motif membuat metode klasifikasi konvensional menjadi kurang efektif dan tidak akurat. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan metode *Convolutional Neural Network*, yang merupakan salah satu pendekatan deep learning unggulan dalam bidang pengolahan citra digital. CNN memiliki kemampuan dalam mengekstraksi fitur visual secara otomatis dan mengenali pola kompleks tanpa perlu proses pra-pemrosesan manual yang rumit. Penelitian ini bertujuan membangun sistem klasifikasi otomatis guna mengenali warna kain tenun Lotis dengan tingkat akurasi tinggi dan efisiensi yang baik. Hasil evaluasi menunjukkan bahwa metode *Convolutional Neural Network* (CNN) mampu melakukan klasifikasi warna kain tenun Lotis berdasarkan asal suku dengan cukup baik, meskipun akurasi antar *fold* belum sepenuhnya konsisten. Sistem ini menjadi langkah awal dalam mendukung digitalisasi dan pelestarian kain tenun sebagai warisan budaya lokal. Diharapkan, teknologi ini dapat memperkaya dokumentasi berbasis digital serta menumbuhkan rasa bangga dan kepedulian generasi muda terhadap kain tradisional daerah. Sistem ini diharapkan dapat memperkuat upaya digitalisasi warisan budaya lokal, meningkatkan dokumentasi berbasis teknologi, serta menumbuhkan rasa bangga dan kepedulian generasi muda terhadap kain tradisional daerah.

Kata kunci: *convolutional neural network*, kain tenun lotis, klasifikasi warna, Timor Tengah Selatan.

1. Pendahuluan

Kain tenun Lotis merupakan warisan budaya yang khas dari Kabupaten Timor Tengah Selatan, Nusa Tenggara Timur. Berakar dari tiga suku besar, yaitu Amanuban, Amanatun, dan Mollo, kain ini mencerminkan kekayaan tradisi setempat melalui motif geometris yang unik dan penggunaan bahan alami. Kain tenun Lotis merupakan kain tenun yang dibuat dengan teknik yang sedikit berbeda yaitu diproses dengan cara mengungkit benang pada bagian pembentuk corak menggunakan alat khusus tenun yang disebut *sia Betty*, (2021). Proses pembuatannya, mulai dari pengolahan kapas hingga pewarnaan dengan bahan-bahan alam, menjadi bukti kearifan lokal yang diwariskan secara turun-temurun. Selain memiliki nilai estetika tinggi, kain tenun Lotis juga berperan dalam berbagai aspek sosial dan budaya, seperti busana adat, mas kawin, alat tukar, hingga simbol status dalam masyarakat. Namun, di tengah modernisasi, pelestarian kain tenun Lotis menghadapi berbagai tantangan, terutama dalam hal klasifikasi dan dokumentasi secara digital. Kompleksitas visual

pada kain tenun Lotis menjadi tantangan tersendiri ketika hendak diklasifikasikan secara otomatis. Setiap kain memiliki variasi motif, corak, dan warna yang unik, yang membuat proses klasifikasi menjadi sulit dilakukan dengan metode konvensional. Selain itu, jumlah data citra yang besar serta variasi warna dan tekstur yang tinggi memerlukan pendekatan yang lebih canggih untuk menghasilkan klasifikasi yang akurat. Dalam konteks pelestarian budaya, kemampuan untuk mengenali dan mengkategorikan kain tenun Lotis secara digital menjadi semakin penting, terutama dalam mendukung dokumentasi serta pengenalan kain tradisional ini di era digital. Oleh karena itu, diperlukan solusi teknologi yang mampu mengatasi tantangan tersebut secara efektif.

Deep learning merupakan salah satu pendekatan yang telah terbukti mampu mengatasi permasalahan dalam pengolahan citra, terutama dalam klasifikasi objek yang kompleks. *Deep learning* merupakan salah satu bagian dari berbagai macam metode *machine learning* yang menggunakan *Artificial Neural Networks* (ANN) jenis pembelajaran dalam *deep learning* dapat berupa supervised, semi-

supervised, dan unsupervised Zamachsari & Puspitasari, (2021). *Deep learning* adalah bagian dari pembelajaran mesin yang berkaitan dengan algoritma dimana cara kerja dari algoritma ini meniru struktur dan fungsi otak yang disebut jaringan saraf tiruan Rochmawati et al.,(2021). Dengan kemampuannya dalam melakukan ekstraksi fitur secara otomatis, *deep learning* memungkinkan sistem komputer untuk mengenali pola dan karakteristik visual secara lebih akurat dibandingkan metode konvensional. Algoritma *deep learning* bekerja dengan cara meniru cara kerja otak manusia dalam memproses informasi, sehingga mampu mendeteksi fitur yang rumit tanpa memerlukan ekstraksi fitur secara manual.

Salah satu metode *deep learning* yang banyak digunakan dalam analisis citra adalah *Convolutional Neural Network* (CNN). CNN merupakan salah satu metode dari *machine learning* yang merupakan pengembangan dari *Multi Layer Perceptron* (MLP) yang mana dirancang untuk mengolah atau membuat data dari dua dimensi Putra et al., (2023). Metode CNN banyak digunakan dalam *image processing* karena tingkat akurasi yang tinggi dan lebih baik dalam pengenalan gambar visual Yuliany et al., (2022). Keunggulan utama CNN terletak pada kemampuannya untuk mengklasifikasikan citra dengan tingkat akurasi tinggi. Hal ini disebabkan oleh kemampuannya untuk mengurangi jumlah parameter bebas dan menangani deformasi gambar input, seperti translasi, rotasi, dan skala Sandy Andika Maulana et al., (2023). CNN telah terbukti efektif dalam mengenali pola, warna, dan tekstur dalam berbagai jenis gambar. Salah satu penelitian yang paling mendekati adalah penelitian oleh Bariyah & Arif Rasyidi, n.d. yang menggunakan CNN untuk klasifikasi multi-label pada motif batik. Dalam penelitian tersebut, CNN diterapkan untuk mengenali lebih dari satu motif dalam satu citra batik, dengan tingkat akurasi mencapai 91,41% dari 300 citra uji. Keberhasilan metode ini menunjukkan bahwa CNN mampu mengklasifikasikan motif kain dengan pola yang kompleks dan beragam. Oleh karena itu, penerapan CNN dalam klasifikasi warna kain tenun Lotis diharapkan dapat menjadi pendekatan yang efektif dalam mengatasi tantangan identifikasi warna dan motif yang beragam. Penelitian dalam bidang klasifikasi citra terus berkembang untuk meningkatkan akurasi dalam mengenali pola dan karakteristik tertentu, termasuk warna dan motif kain. Metode seperti *Convolutional Neural Network* (CNN) telah digunakan secara luas untuk mendukung proses ini. Bowo et al.,(2020) mengaplikasikan CNN dalam klasifikasi motif batik Solo menggunakan dataset 2.256 gambar yang terdiri dari tujuh kelas motif. Sistem ini mencapai akurasi pengujian sebesar 95%, membuktikan bahwa CNN efektif untuk mendeteksi pola batik dengan tingkat efisiensi dan akurasi yang tinggi. Masa & Hamdani, (2021) juga mengintegrasikan CNN untuk klasifikasi batik Yogyakarta. Dengan memanfaatkan teknik pra-

pemrosesan seperti median filter dan *sharpening*, penelitian ini mencapai akurasi 100% menggunakan median filter dan 80% dengan *sharpening*, menunjukkan bahwa metode ini mampu meningkatkan performa klasifikasi. Penelitian selanjutnya oleh Siwilopo & Marcos, (2023) membahas klasifikasi buah jeruk menggunakan algoritma *Convolutional Neural Network* (CNN) dan *K-Nearest Neighbor* (KNN) untuk membandingkan tingkat akurasi kedua metode tersebut. Hasilnya menunjukkan bahwa CNN memiliki akurasi yang lebih tinggi dibandingkan KNN dalam mengidentifikasi tingkat kematangan jeruk berdasarkan warna kulit dan tekstur menggunakan pengolahan citra digital. Pendekatan ini dapat diterapkan pada penelitian "Klasifikasi Warna Kain Tenun Lotis Menggunakan *Convolutional Neural Network*" dengan memanfaatkan metode CNN untuk segmentasi warna kain, meningkatkan akurasi klasifikasi pola dan warna kain Lotis.

Secara keseluruhan, penelitian terdahulu menunjukkan bahwa penggunaan metode CNN sangat efektif untuk klasifikasi citra berbasis warna dan motif. Metode ini terbukti mampu memberikan akurasi tinggi dalam mengenali pola yang kompleks, bahkan pada data dengan kemiripan karakteristik. Dengan kemampuan CNN dalam ekstraksi fitur mendalam, metode ini sangat menjanjikan untuk diaplikasikan pada klasifikasi kain tenun Lotis yang memiliki keragaman warna dan pola unik. Optimasi lebih lanjut pada teknik pra-pemrosesan dan penggunaan dataset yang sesuai akan mendukung peningkatan akurasi dan keandalan sistem klasifikasi. Penelitian ini berkontribusi dalam menerapkan metode CNN untuk klasifikasi warna kain tenun Lotis berdasarkan asal suku, yang masih jarang dijadikan fokus penelitian sebelumnya. Kebaruan dari studi ini terletak pada pemanfaatan pendekatan augmentasi data dan evaluasi *k-fold cross-validation* untuk meningkatkan generalisasi model dalam konteks pelestarian budaya lokal berbasis digital.

2. Metode

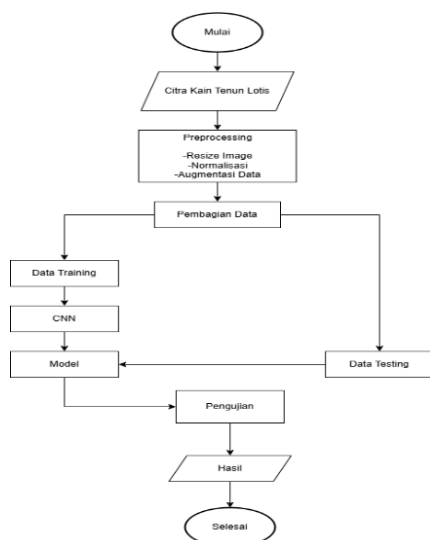
2.1 Ruang Lingkup

Penelitian ini berfokus pada klasifikasi warna kain tenun Lotis menggunakan pendekatan deep learning, khususnya metode *Convolutional Neural Network* (CNN). Cakupan penelitian melibatkan akuisisi data citra kain, pengolahan data, pelatihan model klasifikasi, serta evaluasi dan visualisasi hasil. Dataset yang digunakan terdiri dari 100 citra kain tenun Lotis yang diklasifikasikan berdasarkan asal suku, yaitu 33 citra dari Suku Amanuban, 34 citra dari Suku Amanatun, dan 33 citra dari Suku Mollo. Citra-citra ini dipilih untuk mewakili keragaman warna dan motif khas dari masing-masing suku, sehingga dapat merepresentasikan keunikan kain tenun Lotis secara komprehensif dalam proses klasifikasi.

Pengambilan citra dilakukan secara langsung dengan kamera digital, menggunakan dua teknik pengambilan gambar utama. Teknik pertama adalah *close-up shot* dengan jarak antara 30 hingga 50 cm untuk menangkap detail warna dan tekstur kain secara lebih jelas. Teknik kedua adalah *medium shot* dengan jarak 1 hingga 2 meter untuk memperoleh gambaran umum kain secara keseluruhan. Gambar-gambar yang diambil kemudian disimpan dalam format digital dengan resolusi tinggi dan tanpa kompresi yang berlebihan agar tetap mempertahankan kualitas visual. Seluruh proses klasifikasi dalam penelitian ini dilakukan menggunakan bahasa pemrograman Python. Python merupakan bahasa pemrograman tingkat tinggi yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991 Python juga merupakan bahasa yang sangat populer belakangan ini. Selain itu python juga merupakan bahasa pemrograman yang multi fungsi contohnya python dapat digunakan untuk *Machine Learning* dan *Deep Learning* Alfarizi et al., (2023). Python dipilih karena memiliki ekosistem yang kuat dalam pengembangan model *machine learning* dan *deep learning*, serta didukung oleh berbagai pustaka seperti *TensorFlow* dan *Keras* untuk pembangunan dan pelatihan model CNN, *Scikit-learn* untuk evaluasi model, *OpenCV* untuk pengolahan citra dimana *OpenCV* adalah salah satu *software* pustaka yang ditujukan untuk pengolahan citra dinamis secara *real-time*, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez Zein (2018), serta digunakan *Matplotlib* untuk visualisasi hasil.

2.2 Metodologi

Metode penelitian ini dirancang secara eksperimental dengan menerapkan pendekatan klasifikasi citra berbasis CNN. Tahapan-tahapan yang dilakukan dalam penelitian ini disusun secara sistematis agar dapat direplikasi kembali oleh peneliti lain. Diagram alur yang menggambarkan keseluruhan proses klasifikasi dapat dilihat pada Gambar 1.



Gambar 1. Alur Proses Klasifikasi Warna Kain Tenun Lotis

Langkah pertama adalah akuisisi data, di mana citra kain diperoleh dari lingkungan nyata dengan mempertimbangkan kondisi pencahayaan alami dan buatan guna memastikan kualitas dan konsistensi gambar. Gambar-gambar ini kemudian dimasukkan ke dalam sistem sebagai data mentah untuk diproses lebih lanjut. Tahap berikutnya adalah preprocessing citra. Proses ini dilakukan untuk menyiapkan data agar dapat digunakan secara optimal dalam pelatihan model CNN. Langkah awal dalam *preprocessing* adalah mengubah ukuran (*resize*) seluruh citra ke dalam dimensi yang seragam, yaitu 224×224 piksel. Hal ini dilakukan agar data sesuai dengan format input model CNN yang digunakan. Selanjutnya, citra dinormalisasi dengan membagi seluruh nilai piksel dengan 255 sehingga setiap nilai berada pada rentang 0 hingga 1. Selain itu, dilakukan augmentasi data dengan teknik seperti rotasi, flipping horizontal dan vertikal, *zoom*, serta *brightness adjustment* untuk meningkatkan keragaman data latih dan mengurangi risiko *overfitting* pada model.

Setelah proses *preprocessing* selesai, dataset dibagi menjadi dua subset, yaitu data pelatihan (*training set*) sebanyak 80% dan data pengujian (*testing set*) sebanyak 20%. Pembagian dilakukan secara acak namun stratifikatif, dengan memastikan proporsi representasi dari masing-masing kelas tetap seimbang pada kedua subset tersebut.

Selanjutnya, dilakukan pembangunan dan pelatihan model CNN. Arsitektur model CNN yang digunakan terdiri dari beberapa lapisan utama, yaitu: lapisan konvolusi (*Conv2D*) dengan fungsi aktivasi ReLU untuk mengekstraksi fitur visual dari citra, lapisan *pooling* (*MaxPooling2D*) untuk mereduksi dimensi data dan mempertahankan fitur penting, lapisan *flatten* untuk mengubah data multidimensi menjadi vektor satu dimensi, serta lapisan *fully connected* (*Dense*) yang diakhiri dengan fungsi aktivasi *softmax* pada *layer output*. *Layer output* terdiri dari tiga *neuron*, masing-masing merepresentasikan satu kelas suku: Amanuban, Amanatun, dan Mollo. Model dikompilasi menggunakan optimizer Adam dan fungsi *loss categorical cross-entropy*. Proses pelatihan dilakukan selama 50 *epoch* dengan ukuran *batch* 16. Setelah model selesai dilatih, dilakukan pengujian terhadap data uji yang telah dipisahkan sebelumnya. Hasil pengujian dievaluasi menggunakan metrik performa seperti akurasi, sensitivitas (*recall*), spesifisitas, dan *confusion matrix*. Evaluasi ini penting untuk mengetahui kemampuan model dalam mengklasifikasikan citra kain berdasarkan kelas yang benar. Visualisasi hasil klasifikasi dilakukan melalui grafik *pie chart* untuk menunjukkan distribusi prediksi per kelas, serta *confusion matrix* untuk menilai tingkat kesalahan dan kebenaran prediksi pada masing-masing kelas.

Seluruh proses mulai dari pengambilan data, preprocessing, pelatihan model, pengujian, hingga

evaluasi dan visualisasi dilakukan secara terintegrasi dalam lingkungan pemrograman Python. Hal ini memastikan alur kerja yang efisien, reproduktif, dan mudah disesuaikan dengan eksperimen atau dataset tambahan di masa mendatang.

3. Hasil dan Pembahasan

3.1 Implementasi Sistem

Implementasi sistem dalam penelitian ini dilakukan dengan menggunakan model *Convolutional Neural Network* (CNN) untuk mengklasifikasikan warna kain tenun Lotis berdasarkan motif dan coraknya. Sistem ini dikembangkan menggunakan bahasa pemrograman Python dengan pustaka *TensorFlow* dan *OpenCV*. Dimana *TensorFlow* adalah platform *open source end-to-end* untuk *machine learning* yang dibuat oleh tim *Google Brain*. *TensorFlow* memiliki perkakas, pustaka, dan sumber daya komunitas yang komprehensif dan fleksibel yang memungkinkan para peneliti dalam pengembangan *machine learning* dan tim pengembang dapat dengan mudah membangun dan menerapkan aplikasi yang didukung *machine learning* Arifianto & Muhimmah, (2021). Implementasi melibatkan beberapa tahap utama, yaitu pengumpulan dan pengolahan data citra, pelatihan model CNN, validasi menggunakan metode *k-fold cross validation*, serta pengujian sistem untuk menilai kinerja model. Dataset yang digunakan berasal dari kain tenun Lotis yang dikategorikan berdasarkan tiga suku utama di Timor Tengah Selatan, yaitu Amanuban, Amanatun, dan Mollo. Proses implementasi bertujuan untuk mengoptimalkan akurasi klasifikasi dan meningkatkan efisiensi dalam pengenalan warna serta motif kain tenun Lotis.

3.2 Data Citra

Pada penelitian ini menggunakan data citra kain tenun Lotis dari tiga suku di Timor Tengah Selatan, terdiri dari 100 gambar kain tenun Lotis dengan masing-masing suku diambil sekitar 33-34 sampel. Pengambilan gambar dilakukan dengan jarak kamera 30-50 cm untuk memastikan kualitas citra yang optimal. Gambar disimpan dalam format file *.jpg. Selanjutnya, citra diinput ke dalam Python untuk diproses lebih lanjut menggunakan algoritma CNN.

3.3 Praproses Citra

Pada tahap praproses citra, dilakukan beberapa langkah penting untuk memastikan bahwa data yang digunakan dalam pelatihan model memiliki kualitas yang optimal. Citra asli kain tenun Lotis yang berukuran 675×900 piksel terlebih dahulu diubah ukurannya menjadi 256×256 piksel. Proses *resize* ini bertujuan untuk menyamakan dimensi input model, mengurangi beban komputasi, serta mempercepat proses pelatihan tanpa menghilangkan pola utama kain. *Resize* citra juga merupakan teknik untuk

merubah ukuran citra menjadi lebih besar atau lebih kecil dari citra Pangestu et al., (2020). Setelah itu, citra dikonversi ke *grayscale* guna menyederhanakan informasi warna dan mengurangi kompleksitas data, sehingga model dapat lebih fokus dalam mengenali pola tekstur kain tanpa gangguan variasi warna.

Selain proses dasar seperti *resize* dan *grayscale*, dilakukan juga augmentasi data untuk meningkatkan variasi dalam dataset dan membuat model lebih tangguh dalam mengenali pola kain dari berbagai perspektif. Augmentasi data adalah strategi yang memungkinkan praktisi untuk secara signifikan meningkatkan keragaman data yang tersedia untuk model pelatihan, tanpa benar-benar mengumpulkan data baru. Teknik augmentasi data seperti *cropping*, *padding*, dan *flipping horizontal* umumnya digunakan untuk melatih jaringan neural besar Sanjaya & Ayub, (2020). Salah satu teknik augmentasi yang digunakan adalah rotasi citra pada sudut 45°, 90°, 135°, dan 180°. Dengan melakukan rotasi ini, model diharapkan mampu mengenali motif kain tenun Lotis dalam berbagai orientasi yang mungkin terjadi di dunia nyata, sehingga dapat meningkatkan akurasi dalam proses klasifikasi. Teknik ini tidak hanya memperkaya jumlah data pelatihan, tetapi juga membantu model dalam memahami pola kain dengan lebih baik, menjadikannya lebih adaptif terhadap berbagai kemungkinan variasi dalam citra input.

3.4 Pelatihan Model

Setelah tahap praproses citra selesai, langkah berikutnya adalah pelatihan model menggunakan *Convolutional Neural Network* (CNN). CNN dipilih karena kemampuannya dalam mengenali pola visual yang kompleks, seperti motif dan tekstur pada kain tenun Lotis. Model ini dirancang untuk mengklasifikasikan warna kain tenun Lotis ke dalam tiga kategori utama, yaitu Amanuban, Amanatun, dan Mollo. Model CNN yang dikembangkan dalam penelitian ini terdiri dari beberapa lapisan utama. Lapisan pertama adalah *Convolutional Layer*, yang berfungsi untuk mengekstrak fitur penting dari citra kain tenun menggunakan filter berukuran 3×3. Setelah fitur diekstraksi, aktivasi ReLU (*Rectified Linear Unit*) diterapkan untuk meningkatkan kemampuan model dalam mengenali pola non-linear, ReLU atau *rectified linear unit* merupakan salah satu dari fungsi aktivasi. Fungsi dari ReLU yaitu untuk menghilangkan nilai negatif pada citra. Cara kerja fungsi aktivasi ReLU ini yaitu dengan mengganti nilai negatif pada citra atau *feature maps* dengan nilai 0 Pangestu et al., (2020). Selanjutnya, *Max Pooling* (2×2) digunakan untuk mengurangi dimensi fitur, sehingga meningkatkan efisiensi komputasi dan mengurangi kemungkinan *overfitting*.

Setelah beberapa lapisan konvolusi dan *pooling*, fitur yang telah diproses diubah menjadi bentuk vektor menggunakan *Flattening Layer*. Vektor ini kemudian diteruskan ke *Fully Connected Layer*, yang

menghubungkan fitur yang telah diekstraksi dengan kategori keluaran. Pada bagian terakhir, model menggunakan *Output Layer* dengan aktivasi *Softmax*, yang bertugas menghasilkan probabilitas untuk setiap kelas warna kain tenun Lotis. Fungsi aktivasi *Softmax* dipilih karena model ini melakukan klasifikasi multi-kelas.

Pelatihan model dilakukan dengan algoritma optimasi Adam, yang dipilih karena kemampuannya dalam menyesuaikan laju pembelajaran dan mempercepat proses konvergensi. Fungsi *categorical cross-entropy* digunakan sebagai *loss function*, karena klasifikasi ini terdiri dari tiga kelas yang saling eksklusif. Fungsi *categorical cross-entropy* adalah fungsi *loss (loss function)* yang umum digunakan dalam masalah klasifikasi multi-kelas, di mana setiap sampel hanya dapat dimiliki oleh satu dari beberapa kelas yang saling eksklusif. Model dilatih selama 50 *epoch* dengan *batch size* 32, serta validasi menggunakan *k-fold cross-validation* untuk meningkatkan generalisasi model. Implementasi model CNN dalam penelitian ini dilakukan menggunakan bahasa Python, dengan pustaka Keras dan *TensorFlow*. Struktur kode pelatihan model CNN yang digunakan dapat dilihat pada Gambar 2.

```
pro > modelcnn.py ...
1 import tensorflow as tf
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
4 from tensorflow.keras.preprocessing.image import ImageDataGenerator
5
6 # Inisialisasi model CNN
7 model = Sequential([
8     Conv2D(32, (3, 3), activation='relu', input_shape=(256, 256, 3)),
9     MaxPooling2D(pool_size=(2, 2)),
10
11     Conv2D(64, (3, 3), activation='relu'),
12     MaxPooling2D(pool_size=(2, 2)),
13
14     Conv2D(128, (3, 3), activation='relu'),
15     MaxPooling2D(pool_size=(2, 2)),
16
17     Flatten(),
18     Dense(128, activation='relu'),
19     Dropout(0.5),
20     Dense(3, activation='softmax') # 3 kelas untuk kain tenun Lotis
21 ])
22
23 # Kompilasi model
24 model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
25
26 # Ringkasan arsitektur model
27 model.summary()
```

Gambar 2. Code Implementasi Model CNN

Setelah model dikompilasi, tahap selanjutnya adalah proses pelatihan menggunakan dataset kain tenun Lotis. Dataset ini telah melalui berbagai teknik augmentasi, seperti rotasi, *flipping*, *brightness adjustment*, dan *zooming*, guna meningkatkan akurasi model. Teknik augmentasi ini membantu model mengenali pola kain dari berbagai sudut dan kondisi pencahayaan yang berbeda, sehingga model menjadi lebih tangguh terhadap variasi data.

Model ini diuji pada data latih (80%) dan data uji (20%) yang telah dibagi sebelumnya. Dengan pendekatan ini, model dapat belajar dengan lebih optimal dan memiliki kemampuan generalisasi yang lebih baik saat mengklasifikasikan kain tenun Lotis. Setelah proses pelatihan selesai, arsitektur akhir dari

model CNN yang terbentuk ditampilkan dalam Gambar 3, yang menunjukkan jumlah lapisan, parameter, dan struktur data yang mengalir dalam jaringan.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_1 (Conv2D)	(None, 125, 125, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_2 (Conv2D)	(None, 60, 60, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 128)	0
flatten (Flatten)	(None, 115200)	0
dense (Dense)	(None, 128)	14,745,728
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387
Total params: 14,839,363 (56.61 MB)		
Trainable params: 14,839,363 (56.61 MB)		
Non-trainable params: 0 (0.00 B)		

Gambar 3. Output Model CNN

Model ini terdiri dari beberapa lapisan, dimulai dengan tiga lapisan *Conv2D* (konvolusi) yang masing-masing diikuti oleh lapisan *MaxPooling2D* untuk pengurangan dimensi spasial. Lapisan pertama menghasilkan 32 filter, lapisan kedua 64 filter, dan lapisan ketiga 128 filter, dengan output shape yang semakin mengecil karena pooling. Setelah itu, data diratakan menggunakan lapisan *Flatten*, menghasilkan vektor berdimensi 115200. Lapisan ini diikuti oleh lapisan *Dense* dengan 128 unit, kemudian lapisan *Dropout* untuk mencegah *overfitting*, dan akhirnya lapisan *Dense* terakhir dengan 3 unit output, yang kemungkinan digunakan untuk klasifikasi tiga kelas. Model ini memiliki total parameter sebanyak 14.839.363, semuanya dapat dilatih (*trainable*), tanpa parameter *non-trainable*. Ukuran model ini sekitar 56,61 MB. Selama pelatihan 50 *epoch*, model menunjukkan peningkatan akurasi secara bertahap. Selama pelatihan model menunjukkan perkembangan akurasi dan *loss* yang terekam dalam grafik hasil pelatihan. Rangkuman hasil pelatihan selama 50 *epoch* dapat dilihat pada Gambar 4. Berikut adalah output dari proses pelatihan model:

```
Epoch 50/50
3/3 30s 7s/step - accuracy: 0.752 - loss: 0.5673 - val_accuracy: 0.4444 - val_loss: 1.5399
```

Gambar 4. Output Proses Pelatihan Model

Output tersebut merupakan hasil dari proses pelatihan model *machine learning* yang telah mencapai *epoch* ke-50 dari total 50 *epoch*. Proses pelatihan menggunakan 3 batch data, membutuhkan waktu 30 detik dengan rata-rata 7 detik per step. Pada data pelatihan, model mencapai akurasi 72.52% dengan *loss* sebesar 0.5673. Namun, pada data validasi, akurasi model hanya mencapai 44.44% dengan *loss* yang lebih tinggi, yaitu 1.5399. Perbedaan signifikan antara akurasi dan *loss* pada

data pelatihan dan validasi mengindikasikan adanya potensi *overfitting*, di mana model terlalu baik dalam mempelajari data pelatihan namun kurang mampu menggeneralisasi pada data baru.

K-Fold Cross Validation

Setelah model *Convolutional Neural Network* (CNN) dilatih, tahap berikutnya adalah melakukan validasi untuk mengevaluasi performa model sebelum dilakukan pengujian lebih lanjut. Salah satu teknik validasi yang digunakan dalam penelitian ini adalah *K-Fold Cross Validation*. *K-Fold CV* adalah kumpulan data yang diberikan dibagi menjadi sejumlah *K* bagian / *fold* di mana setiap *fold* digunakan sebagai set pengujian di beberapa titik Peryanto et al., (2020). *K-fold cross validation* merupakan salah satu dari teknik yang difungsikan untuk memilah data menjadi *train* data serta test data. Teknik ini diterapkan karena dapat mengurangi bias yang didapatkan didalam pengambilan sebuah sampel Ridwansyah, (2022). Teknik ini digunakan untuk mengurangi bias akibat pembagian dataset yang tidak merata serta memastikan bahwa model dapat melakukan generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya.

Dalam penelitian ini, digunakan *5-Fold Cross Validation* ($K=5$), yang berarti dataset dibagi menjadi 5 bagian yang sama besar. Pada setiap iterasi pelatihan, 4 bagian digunakan untuk melatih model, sementara 1 bagian digunakan untuk menguji model. Proses ini diulang sebanyak 5 kali, sehingga setiap bagian dataset berkesempatan menjadi data uji. Nilai rata-rata dari hasil validasi digunakan sebagai tolak ukur kinerja model secara keseluruhan. Setelah model diuji pada 5 *fold*, evaluasi performa model dilakukan menggunakan metode *5-fold cross validation* untuk menguji kemampuan generalisasi CNN. Hasil pelatihan pada setiap *fold* dirangkum dalam Tabel 1 hingga Tabel 5 berikut.

Tabel 1. Training k-fold 1

Epoch	Batch size	Accuracy (%)	Loss
50	16	40.00	2.7709

Pada *fold* pertama, model mencapai akurasi 40% dengan nilai *loss* 2.7709. Ini menunjukkan model cukup baik dalam mengenali pola pada data *fold* 1, meskipun masih ada ruang untuk perbaikan.

Tabel 2. Training k-fold 2

Epoch	Batch size	Accuracy (%)	Loss
50	16	15.00	9.3115

Pada *fold* kedua, akurasi turun drastis menjadi 15% dan *loss* meningkat tajam ke 9.3115. Ini menandakan model mengalami kesulitan dalam mempelajari data pada *fold* ini, kemungkinan karena distribusi data yang berbeda atau data yang lebih sulit.

Tabel 3. Training k-fold 3

Epoch	Batch size	Accuracy (%)	Loss
50	16	35.00	3.0972

Pada *fold* ketiga, akurasi naik menjadi 35% dengan *loss* 3.0972. Hasil ini lebih baik dibanding *fold* kedua, namun masih di bawah *fold* pertama.

Tabel 4. Training k-fold 4

Epoch	Batch size	Accuracy (%)	Loss
50	16	25.00	3.7751

Pada *fold* keempat, akurasi berada di angka 25% dengan *loss* 3.7751. Model menunjukkan performa sedang, tidak terlalu baik maupun terlalu buruk.

Tabel 5. Training k-fold 5

Epoch	Batch size	Accuracy (%)	Loss
50	16	35.00	8.5922

Pada *fold* kelima, akurasi kembali ke 35%, namun *loss* cukup tinggi di angka 8.5922. Ini menunjukkan model bisa menebak dengan benar pada beberapa data, tetapi prediksi yang salah memiliki error yang besar.

Dalam konteks klasifikasi warna kain tenun Lotis, penggunaan *5-Fold Cross Validation* bertujuan untuk memastikan bahwa model tidak hanya belajar dari data tertentu saja, tetapi juga mampu mengenali pola dari keseluruhan dataset secara menyeluruh. Dengan membagi dataset menjadi lima bagian yang berbeda, setiap bagian akan berperan sebagai data uji setidaknya satu kali, sehingga mengurangi kemungkinan model mengalami *overfitting* atau bias terhadap data tertentu. Proses ini memberikan evaluasi performa yang lebih adil dan menyeluruh dibandingkan pembagian data satu kali (*train-test split*).

Selain itu, teknik ini memungkinkan pengujian stabilitas dan konsistensi kinerja model. Jika performa model konsisten di kelima *fold*, maka dapat disimpulkan bahwa model memiliki generalisasi yang baik terhadap data baru. Sebaliknya, jika variasi hasil antar *fold* cukup besar, hal ini menjadi indikasi bahwa model belum optimal atau data masih belum cukup representatif. Oleh karena itu, hasil dari *5-Fold Cross Validation* tidak hanya memberikan rata-rata akurasi, tetapi juga menjadi dasar untuk mengevaluasi kekuatan dan kelemahan model secara lebih mendalam. Evaluasi ini sangat penting terutama dalam pengembangan sistem klasifikasi berbasis citra yang kompleks seperti kain tenun Lotis.

4. Kesimpulan

Penelitian ini membuktikan bahwa metode *Convolutional Neural Network* (CNN) mampu digunakan untuk mengklasifikasikan warna kain tenun Lotis berdasarkan asal suku dengan cukup baik, meskipun hasil akurasi antar *fold* belum konsisten. Sistem ini menjadi langkah awal dalam mendukung digitalisasi dan pelestarian kain tenun sebagai warisan budaya. Ke depan, penelitian dapat dikembangkan dengan menggunakan dataset yang lebih besar dan beragam, menerapkan transfer

learning untuk efisiensi pelatihan, serta memanfaatkan citra berwarna agar informasi visual lebih kaya. Selain itu, pengembangan sistem klasifikasi berbasis aplikasi *mobile* atau web juga dapat menjadi arah konkret untuk penerapan teknologi ini secara luas.

Daftar Pustaka:

- Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning. *Karya Ilmiah Mahasiswa Bertauhid (KARIMAH TAUHID)*, 2(1), 1–6.
- Arifianto, J., & Muhimmah, I. (2021). Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Algoritma Deep Learning dengan TensorFlow. *Journal Automata*, 21–29.
- Bariyah, T., & Arif Rasyidi, M. (n.d.). Convolutional Neural Network Untuk Metode Klasifikasi Multi-Label Pada Motif Batik Convolutional Neural Network for Multi-Label Batik Pattern Classification Method. In *Februari* (Vol. 20, Issue 1).
- Betty, N. A. (2021). *Futus, Lotis dan Buna: Makna Teologis dan Peran Etika Sosial Tenun Ikat Amanuban Barat dalam membentuk harmoni masyarakat Desa Tubuhue-TTS*. 1–33. <https://doi.org/https://repository.uksw.edu/handle/123456789/35096>
- Bowo, T. A., Syaputra, H., & Akbar, M. (2020). Penerapan Algoritma Convolutional Neural Network Untuk Klasifikasi Motif Citra Batik Solo. *Journal of Software Engineering Ampera*, 1(2), 82–96. <https://doi.org/10.51519/journalsea.v1i2.47>
- Masa, A. P. A., & Hamdani, H. (2021). Klasifikasi Motif Citra Batik Menggunakan Convolutional Neural Network Berdasarkan K-means Clustering. *Jurnal Media Informatika Budidarma*, 5(4), 1292. <https://doi.org/10.30865/mib.v5i4.3246>
- Pangestu, A. R., Basuki Rahmat, & Fetty Tri Anggraeny. (2020). Implementasi Algoritma Cnn Untuk Klasifikasi CitraLahan Dan Perhitungan Luas. *Jurnal Informatika Dan Sistem Informasi (JIFoSI)*, 1(1), 166–174.
- Peryanto, A., Yudhana, A., & Umar, R. (2020). Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation. *Journal of Applied Informatics and Computing*, 4(1), 45–51. <https://doi.org/10.30871/jaic.v4i1.2017>
- Putra, J. V. P., Ayu, F., & Julianto, B. (2023). Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN. *Stains (Seminar Nasional Teknologi & Sains)*, 2(1), 155–162. <https://proceeding.unpkediri.ac.id/index.php/stains/article/view/2888>
- Ridwansyah, T. (2022). Implementasi Text Mining Terhadap Analisis Sentimen Masyarakat Dunia Di Twitter Terhadap Kota Medan Menggunakan K-Fold Cross Validation Dan Naïve Bayes Classifier. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, 2(5), 178–185. <https://doi.org/10.30865/klik.v2i5.362>
- Rochmawati, N., Hidayati, H. B., Yamasari, Y., Tjahyaningtijas, H. P. A., Yustanti, W., & Prihanto, A. (2021). Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam. *Journal of Information Engineering and Educational Technology*, 5(2), 44–48. <https://doi.org/10.26740/jieet.v5n2.p44-48>
- Sandy Andika Maulana, Shabrina Husna Batubara, Tasya Ade Amelia, & Yohanna Permata Putri Pasaribu. (2023). Penerapan Metode CNN (Convolutional Neural Network) Dalam Mengklasifikasi Jenis Ubur-Ubur. *Jurnal Penelitian Rumpun Ilmu Teknik*, 2(4), 122–130. <https://doi.org/10.55606/juprit.v2i4.3084>
- Sanjaya, J., & Ayub, M. (2020). Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup. *Jurnal Teknik Informatika Dan Sistem Informasi*, 6(2), 311–323. <https://doi.org/10.28932/jutisi.v6i2.2688>
- Siwilopo, K. P., & Marcos, H. (2023). Membandingkan Klasifikasi Pada Buah Jeruk Menggunakan Metode Convolutional Neural Network Dan K-Nearest Neighbor. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*, 12(1), 57–64. <https://doi.org/10.34010/komputa.v12i1.9068>
- Yuliany, S., Aradea, & Andi Nur Rachman. (2022). Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Buana Informatika*, 13(1), 54–65. <https://doi.org/10.24002/jbi.v13i1.5022>
- Zamachsari, F., & Puspitasari, N. (2021). Penerapan Deep Learning dalam Deteksi Penipuan Transaksi Keuangan Secara Elektronik. *Jurnal RESTI*, 5(2), 203–212. <https://doi.org/10.29207/resti.v5i2.2952>
- Zein, A. (2018). Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON. *Sainstech: Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi*, 28(2), 22–26. <https://doi.org/10.37277/stch.v28i2.238>

Halaman ini sengaja dikosongkan