

Modifikasi Arsitektur dalam *Convolutional Neural Network* untuk Klasifikasi Batik Lampung dan Batik Yogyakarta

Yonathan Octavianus ^{1*}, Donny Avianto ²

^{1,2} Program Studi Informatika, Fakultas Sains & Teknologi, Universitas Teknologi Yogyakarta, Kabupaten Sleman, Provinsi Daerah Istimewa Yogyakarta, Indonesia.

Corresponding Email: yonathanoctavianus111002@gmail.com ^{1*}

Histori Artikel:

Dikirim 2 Desember 2024; *Diterima dalam bentuk revisi* 15 Desember 2024; *Diterima* 1 Januari 2025; *Diterbitkan* 10 Januari 2025. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STMIK Indonesia Banda Aceh.

Abstrak

Batik merupakan salah satu bentuk keunikan budaya Indonesia. Pada tanggal 2 Oktober 2009 UNESCO (United Nations Educational, Scientific, and Cultural Organization) menetapkan batik sebagai warisan kemanusiaan untuk budaya lisan dan non-bendawi (Masterpiece of the Oral and Intangible Heritage of Humanity). Salah satu dari batik warisan nenek moyang kita adalah Batik Lampung dan Batik Yogyakarta yang Dimana kedua batik tersebut memiliki perbedaan dan keunikan masing-masing, sehingga kita sebagai masyarakat Indonesia harus menjaga dan melestarikan warisan budaya dari nenek moyang kita dengan membuat sebuah sistem yang dapat menentukan kedua batik tersebut tanpa menggunakan naluri ataupun berdasarkan rekomendasi orang lain yang masih bisa menyebabkan kesalahan. Pada penelitian sebelumnya, klasifikasi menggunakan Multikernel SVM berhasil mencapai akurasi sebesar 100%. Ada pula yang menggunakan CNN(Convolutional Neural Network)-Sobel dengan akurasi sebesar 91,2% pada proses training dan 91,8% pada proses validation. Permasalahan yang di alami dalam penelitian sebelumnya adalah adanya keterbatasan pada dataset dan proses pengujian model yang masih belum maksimal sehingga belum mendapatkan hasil yang memuaskan sehingga pada penelitian ini akan menggunakan metode Convolutional Neural Network dengan 6 arsitektur yang 3 di antaranya arsitektur yang tidak dimodifikasi yaitu MobileNetV2, DenseNet121, dan Xception. Serta 3 arsitektur yang dimodifikasi yaitu MobileNetV2 (Modified), DenseNet121 (Modified), dan Xception (Modified). Pemilihan ke tiga arsitektur tersebut karena memiliki lapisan layer yang sangat banyak sehingga dapat mengitung data yang jumlahnya sangat banyak dan menghasilkan output yang sesuai. Hasil terbaik yang didapatkan pada penelitian ini adalah arsitektur yang dimodifikasi yaitu Xception (Modified) dengan akurasi sebesar 100%, Presisi 97%, Recall 94%. F1 Score 92%, dan Loss 0,0066 pada eksperimen epoch ke 30 dan learning rate 0,0001 sehingga Xception menjadi model terbaik pada arsitektur yang dimodifikasi (Modified). Penelitian ini di harapkan mampu memberikan sistem teknologi terbaru kepada orang awam yang tidak mengenal Batik Lampung dan Batik Yogyakarta untuk dapat membedakan diantara keduanya secara spesifik sehingga meminimalisir adanya kesalahan dalam menganalisa atau ketika membeli Batik yang di inginkan.

Kata Kunci: Convolutional Neural Network; Modifikasi; Batik Lampung; Batik Yogyakarta; Arsitektur.

Abstract

Batik is one of the unique forms of Indonesian culture. On October 2, 2009 UNESCO (United Nations Educational, Scientific, and Cultural Organization) designated batik as a Masterpiece of the Oral and Intangible Heritage of Humanity. One of the batik heritages of our ancestors is Lampung Batik and Yogyakarta Batik, where both batik have their own differences and uniqueness, so that we as Indonesian people must maintain and preserve the cultural heritage of our ancestors by creating a system that can determine both batik without using instinct or based on recommendations from others which can still cause errors. In previous studies, classification using Multikernel SVM managed to achieve an accuracy of 100%. There are also those who use CNN(Convolutional Neural Network)-Sobel with an accuracy of 91,2% in the training process and 91,8% in the validation process. The problems experienced in previous studies were the limitations of the dataset and the model testing process which was still not optimal so that it did not get satisfactory results so that in this study the Convolutional Neural Network method will be used with 6 architectures, 3 of which are unModified architectures, namely MobileNetV2, DenseNet121, and Xception. And 3 Modified architectures, namely MobileNetV2 (Modified), DenseNet121 (Modified), and Xceptipn (Modified). The selection of the three architectures is because it has a very large number of layers so that it can calculate a very large amount of data and produce the appropriate output. The best results obtained in this study were the Modified architecture, namely Xception (Modified) with an accuracy of 100%, Precision 97%, Recall 94%. F1 Score 92%, and Loss 0,0066 in the 30th epoch experiment and learning rate 0,0001 so that Xception became the best model in the Modified architecture (Modified). This research is expected to be able to provide a renewable technology system to ordinary people who do not know Lampung Batik and Yogyakarta Batik to be able to distinguish between the two specifically so as to minimize errors in analyzing or when buying the desired Batik

Keyword: Convolutional Neural Network; Modification; Lampung Batik; Yogyakarta Batik; Architecture.

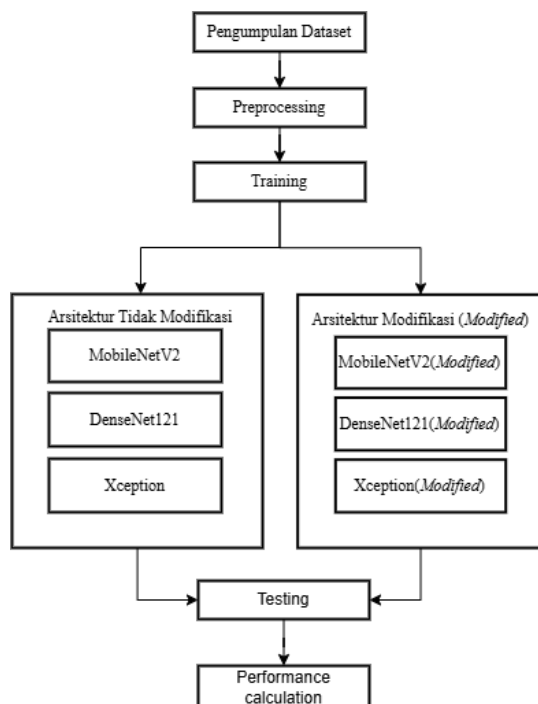
1. Pendahuluan

Batik merupakan salah satu bentuk keunikan budaya Indonesia yang kaya akan makna dan nilai estetika. Pada tanggal 2 Oktober 2009, UNESCO (*United Nations Educational, Scientific, and Cultural Organization*) menetapkan batik sebagai Warisan Kemanusiaan untuk Budaya Lisan dan Non-Bendawi (*Masterpiece of the Oral and Intangible Heritage of Humanity*). Penetapan ini bukan hanya mengakui keindahan batik, tetapi juga menegaskan pentingnya pelestarian budaya yang menjadi warisan turun-temurun. Oleh karena itu, pelestarian batik menjadi kewajiban kita sebagai generasi penerus (Raharja & Widyatusti, 2022). Pentingnya memahami sejarah batik tidak dapat dipandang sebelah mata. Tanpa pemahaman yang mendalam mengenai asal-usul dan proses pembuatan batik, kita akan kesulitan untuk menghargai kerumitan dan keindahan seni ini. Sejarah batik mencerminkan perjalanan budaya dan identitas masyarakat Indonesia yang telah ada selama berabad-abad. Setiap motif dan warna dalam batik memiliki makna tersendiri yang berkaitan dengan tradisi dan filosofi masyarakat setempat (Atmaja & Lusiana, 2023). Namun, dalam praktiknya, penentuan jenis batik sering kali masih mengandalkan firasat atau rekomendasi dari orang lain. Hal ini dapat menyulitkan pengenalan ciri khas batik yang sebenarnya. Banyak kasus menunjukkan bahwa pembeli Batik Lampung dan Batik Yogyakarta merasa dirugikan akibat hanya mengandalkan rekomendasi verbal, sehingga batik yang dibeli tidak sesuai dengan harapan dan kualitas yang diinginkan (Ayu Ratna Juwita *et al.*, 2021). Keterbatasan informasi dan pengetahuan tentang batik dapat mengakibatkan kesalahan dalam memilih dan menghargai karya seni ini.

Berbagai penelitian tentang klasifikasi batik telah dilakukan sebelumnya untuk membantu dalam pengenalan dan pemahaman batik. Andono & Rachmawanto (2021) menggunakan *Multikernel SVM* dan *Ekstraksi Fitur GLCM* dan *LBP*, dengan hasil akurasi terbaik mencapai 100% pada kernel polinomial. Utari & Zulfikar (2023) melakukan klasifikasi batik menggunakan *CNN (Convolutional Neural Network)-Canny* dan *CNN-Sobel*, di mana akurasi tertinggi diperoleh oleh *CNN-Sobel* dengan hasil 91,2% pada proses pelatihan dan 91,8% pada proses validasi. Penelitian lain oleh Masa & Hamdani (2021) juga berhasil melakukan klasifikasi motif batik menggunakan *CNN* berdasarkan *K-means Clustering*, yang menghasilkan akurasi 100%. Selain itu, proses klasifikasi dari hasil sharpening mencapai akurasi sebesar 80% (Yanuardi *et al.*, 2024). Meskipun banyak penelitian telah menunjukkan hasil yang menjanjikan, masih ada tantangan yang perlu diatasi. Keterbatasan pada dataset dan proses pengujian model yang belum optimal menjadi tantangan dalam penelitian sebelumnya (Hawibowo & Muhimmah, 2024). Oleh karena itu, penelitian ini bertujuan untuk mencoba klasifikasi batik menggunakan tiga arsitektur: *MobileNetV2*, *DenseNet121*, dan *Xception*, yang akan dimodifikasi untuk meningkatkan hasil akurasi. Keunggulan dari ketiga arsitektur yang dimodifikasi adalah stabilitas hasil akurasi yang lebih baik, dengan memotong 30 lapisan dan menambahkan 4 lapisan baru pada bagian klasifikasi. Dengan modifikasi ini, diharapkan penelitian ini dapat menghasilkan output yang lebih optimal dan stabil, serta memberikan kontribusi yang berarti bagi pelestarian dan pengenalan batik di Indonesia (Putra *et al.*, 2023).

2. Metode Penelitian

Penelitian ini memiliki 6 tahapan yaitu, Pengumpulan Dataset, lalu *Preprocessing*, Augmentasi, *Training* (Arsitektur Tidak Modifikasi dan Arsitektur Modifikasi), *Testing*, dan menghitung Performa model. Alur Metode Penelitian dapat di tinjau pada gambar 1.



Gambar 1. Model Proses Penelitian

2.1 Pengumpulan Dataset

Sumber data yang digunakan dalam penelitian ini adalah dataset yang berasal dari *kaggle* sebanyak 600 sebagai data latih, dan data primer merupakan data yang di ambil langsung dari lokasi toko Batik Lampung dan Yogyakarta sebagai data uji sebanyak 128. Berikut merupakan contoh dataset Batik Lampung yang di tunjukkan pada gambar 2. (a) dan contoh Batik Yogyakarta yang di tunjukkan pada gambar 2. (b) di bawah ini.



Gambar 2. (a). Batik Lampung; (b). Batik Yogyakarta

2.2 Preprocessing

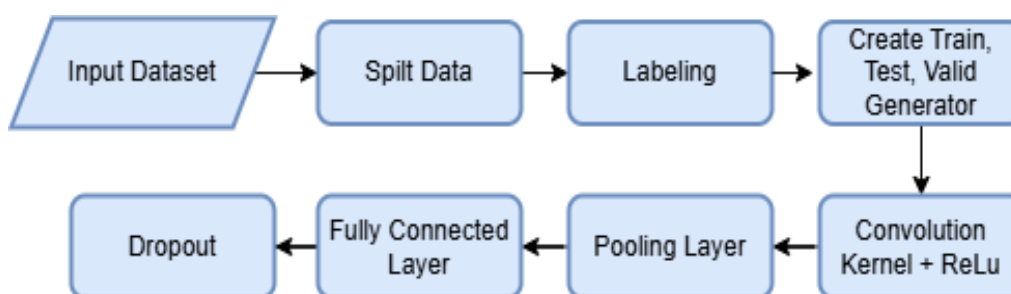
Pre-processing merupakan salah satu tahapan yang digunakan untuk menghilangkan *noise* pada data citra yang akan digunakan Ratna Indah Juwita Harahap *et al.* (2024). *Pre-processing* sangat penting digunakan karena untuk mendukung proses pengolahan yang melibatkan beberapa tahapan yang akan membuat suatu data yang di olah menjadi berkualitas. Pada penelitian ini menggunakan tahapan dalam *Pre-processing* yaitu *resize* yang berfungsi untuk mengubah ukuran pixel asli ke dalam ukuran yang lebih kecil agar lebih cepat dalam pemrosesannya. Dan pada penelitian ini, ukuran gambar yang tadinya berasio aspek 4:3 menjadi 2:1. Hal itu dilakukan agar ukuran gambar tidak terlalu besar yang mengalibatkan proses pelatihan data tidak terlalu banyak memakan waktu, sehingga peneliti memperkecil ukuran gambar untuk mempermudah melakukan proses *Pre-processing*.

2.3 Augmentasi

Proses Augmentasi merupakan tahap dimana dataset mengalami peningkatan jumlah atau memperkaya dataset supaya ketika dilakukan proses training, hasil testing mengalami peningkatan pada akurasi dan validasi tidak mengalami *overfitting* yang berlebihan Nisa & Candra (2023). Augmentasi sangat di perlukan pada penelitian ini karena selain untuk memperkaya dataset, Augmentasi juga di gunakan untuk memperkuat hasil akurasi yang di latih menggunakan model serta mengurangi ketidakseimbangan pada dataset yang kita miliki. Pada penelitian ini, tahapan yang di lakukan pada Augmentasi adalah Scaling, dengan mengubah pixel gambar ke ukuran 255 serta melakukan Rotasi dengan memutar sudut menjadi 15° untuk membuat gambar menjadi simetris dalam skala vertikal. Dengan melakukan augmentasi, dataset yang dimiliki mampu menghasilkan performa model yang lebih optimal karena augmentasi memperkaya dataset dan proses Scaling yang membuat data lebih seimbang dan terukur sehingga meminimalisir adanya *overfitting* pada saat evaluasi model dilakukan Fahrurroji *et al.* (2024).

2.4 Training

Pada proses training merupakan proses pembentukan atau pembuatan model yang melibatkan arsitektur yang akan di gunakan setelah melewati proses *Pre-processing* dan Augmentasi. Karena pada penelitian ini menggunakan 6 Arsitektur, yaitu *MobileNetV2*, *MobileNetV2(Modified)*, *DenseNet121*, *DenseNet121(Modified)*, *Xception*, dan *Xception(Modified)*. Tahapan yang di lalui proses *training* pada penelitian ini adalah *Split data*, *Labeling*, *Create Train, Test Valid Generator*, *Dropout*, *Fully Conncted Layer*, *Pooling Layer*, dan *Convolution Kernel*. Kemudian akan di lakukan beberapa eksperimen dengan mengubah beberapa parameter dan epoch untuk melihat seberapa besar akurasi yang di dapat. Alur di atas akan di perjelas pada Gambar 3 di bawah ini.



Gambar 3. Alur Tahapan *Training*

Pada gambar di atas, merupakan alur tahapan *training* yang dimulai dengan memasukkan dataset, kemudian dilakukan proses olah gambar dengan melakukan pembagian data atau *Split data*. Pada penelitian ini akan melakukan eksperimen dengan membagi data 60:40 untuk arsitektur yang tidak di modifikasi, dan 80:20 untuk arsitektur yang dimodifikasi. Setelah itu masuk ke proses *labelling* yang mengelompokkan dataset menjadi 2 class yaitu Batik Lampung dan Batik Yogyakarta, serta memvalidasi data train dan test menggunakan generator. Setelah proses di atas selesai, selanjutnya masuk ke proses klasifikasi menggunakan CNN yang dimana gambar akan di konvolusi menggunakan aktivasi Relu. Setelah itu konvolusi berlanjut masuk ke dalam *Pooling Layer* untuk dilakukan proses *Max Pooling*. Proses selanjutnya akan dilakukan perkalian feed forward pada fully connected layer. Untuk mengurangi over fitting pada data, dilakukan proses *Dropout* menggunakan aktivasi *softmax*. Setelah semua selesai dilakukan, tahap selanjutnya menyimpan model untuk dilakukan uji coba yang akan di lakukan pada proses *testing* Agustiani *et al.* (2022). Selain itu, di tahap *training* akan di lakukan beberapa eksperimen terhadap parameter pada arsitektur sebelum di modifikasi dan yang di modifikasi di tinjau pada tabel 1 dan 2 di bawah ini.

Tabel 1. Parameter Arsitektur Yang Tidak di Modifikasi

Arsitektur	Parameter	Tipe
<i>MobileNetV2</i>	<i>Epoch</i>	10, 20, 30
<i>DenseNet121</i>	Split Data	80:20
<i>Xception</i>	<i>Rescaling</i>	299
	<i>Optimizer</i>	Adam
	<i>Learning rate</i>	0,01 0,001 0,0001

Tabel 2. Parameter Arsitektur Yang di Modifikasi

Arsitektur	Parameter	Tipe
<i>MobileNetV2</i>	<i>Epoch</i>	10, 20, 30
<i>DenseNet121</i>	Split Data	80:20
<i>Xception</i>	<i>Rescaling</i>	299
	<i>Optimizer</i>	Adam
	<i>Learning rate</i>	0,01 0,001, 0,0001

Dalam proses *training*, Selain menggunakan arsitektur yang telah di sebutkan di atas, penelitian ini juga akan menggunakan arsitektur sebelumnya yang akan di modifikasi dengan mengubah parameter dan mengubah lapisan pada arsitektur. Hal ini dilakukan agar mengetahui seberapa besar hasil yang didapat setelah dilakukan modifikasi. Untuk itu, arsitektur yang semula default akan diberi tambahan kata (*Modified*) yang berarti sudah di ubah atau dimodifikasi dengan memotong 30 layer aslinya dan menambahkan 4 layer baru pada base model masing-masing arsitektur. Sehingga arsitektur yang telah dimodifikasi berubah nama menjadi :

- 1) *MobileNetV2 (Modified)*
- 2) *DenseNet121 (Modified)*
- 3) *Xception (Modified)*

Pembeda antara arsitektur yang tidak dimodifikasi dengan arsitektur yang di modifikasi adalah terletak pada parameter yang berbeda seperti epoch, *learning rate* serta *rescaling* dan split data serta layer yang digunakan pada arsitektur yang dimodifikasi menggunakan layer yang dimodifikasi dengan memotong 30 bagian layer akhir pada arsitektur aslinya kemudian menambahkan 4 layer baru. Hal ini dilakukan untuk mengoptimalkan hasil akurasi yang masih belum stabil pada saat belum dilakukan modifikasi, dengan memotong 30 layer pada bagian base model untuk menggantikan layer sebelumnya dan menggantikannya dengan 4 layer baru yang diharapkan menghasilkan *output* memuaskan yang dapat di tinjau pada tabel 3 di bawah ini.

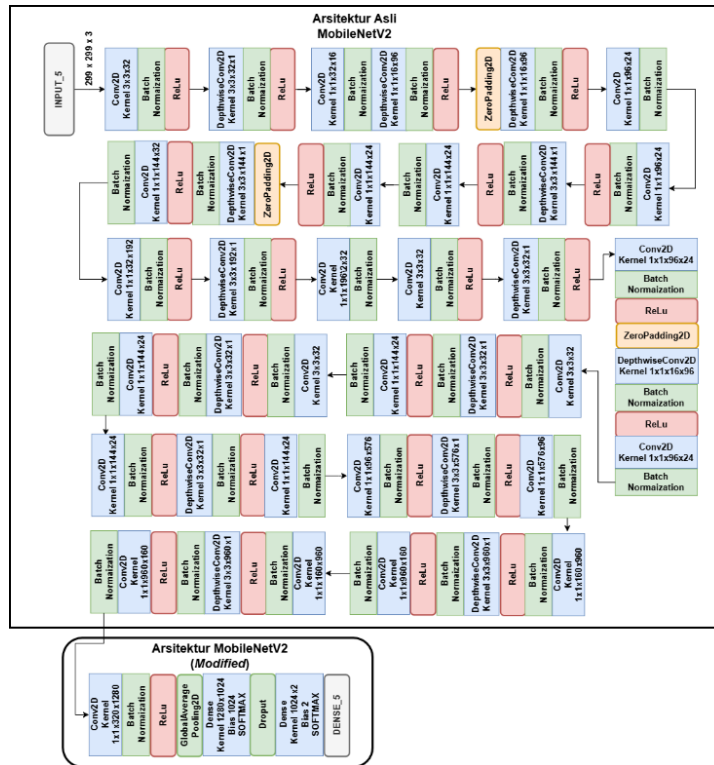
Tabel 3. Layer Arsitektur Modifikasi

Layer Modifikasi
GlobalAveragePooling2D
Dense (1024, Activation='Relu')
Dropout (0,5)
Dense (class_count, Activation ='softmax')

Berikut ini merupakan 3 arsitektur yang telah dimodifikasi (*Modified*) adalah :

- 1) *MobileNetV2 (Modified)*
MobileNetV2 (Modified) merupakan hasil modifikasi dari arsitektur dasar *MobileNetV2* dengan tujuan untuk memperkuat akurasi pada klasifikasi Batik Lampung dan Batik Yogyakarta. Arsitektur ini memiliki sekitar 53 layers dengan lapisan utama convolutional layers, deepwise separable convolutions dan lapisan pendukung lainnya seperti batch normalization dll. Dengan menghilangkan 30 layer terakhir berfungsi untuk jaringan dan mempercepat waktu komputasi. Sebagai pengganti, 4 layer baru di tambahkan di bagian *output* yaitu: *GlobalAveragePooling2D*,

Dense(1024, activation= Relu, Dropout(0.5), dan Dense(class_count, activation=softmax)) yang dapat di tinjau pada tabel 3. Hal ini dilakukan untuk menjangir lebih banyak pola penting pada data sehingga menjadi lebih baik selama proses pelatihan. Arsitektur hasil modifikasi ini dapat di tinjau pada gambar 4.

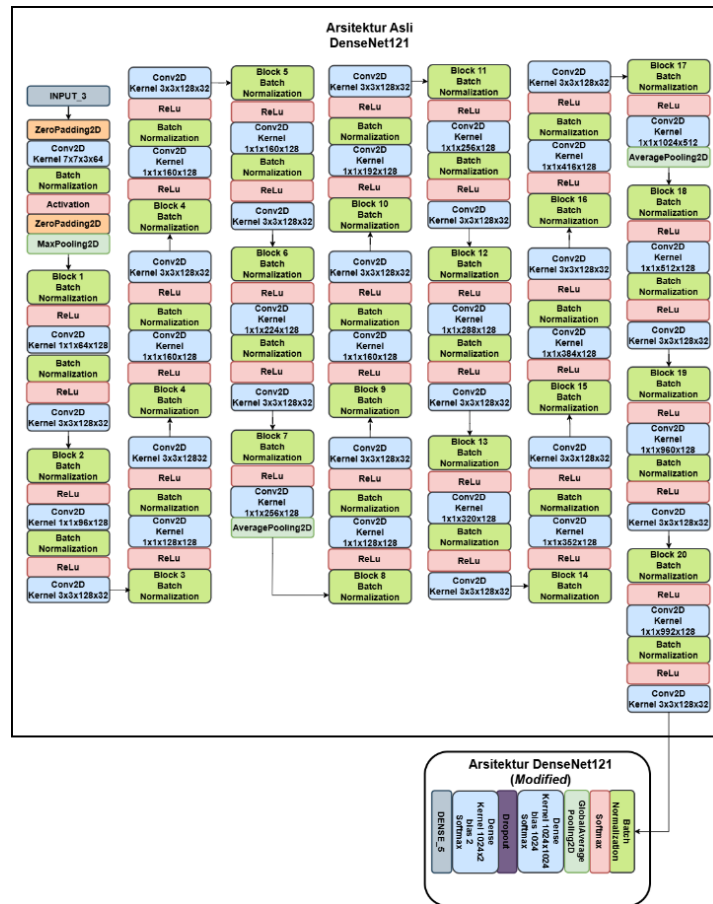


Gambar 4. Arsitektur *MobileNetV2(Modified)*

Meskipun 30 lapisan *MobileNetV2* telah di hilangkan, dengan hadirnya 4 lapisan baru semakin naik jumlah total parameternya. Sebelum modifikasi, total parameter pada *MobileNetV2* sebesar 2,257,984 akan tetapi setelah di lakukan modifikasi berubah menjadi 3,571,778. Hal ini disebabkan oleh penambahan lapisan Dense dengan 1024 neuron, yang memerlukan lebih banyak parameter untuk menghubungkan neuron-neuron dari lapisan sebelumnya. Dengan jumlah parameter yang bertambah, model memiliki kemampuan lebih untuk mengenali pola-pola penting, sehingga bisa meningkatkan hasil klasifikasinya.

2) *DenseNet121(Modified)*

DenseNet121(Modified) adalah hasil dari modifikasi arsitektur sebelumnya yaitu *DenseNet121*. Arsitektur ini memiliki 121 layer yang terdiri dari convolutional layers, pooling layers, dan dense layers. Arsitektur ini menggunakan dense block yang menghubungkan semua layer depan untuk meningkatkan fitur. Sama dengan sebelumnya *DenseNet121(Modified)* ini dimodifikasi dengan memotong 30 layer bagian dan menambah 4 layer baru di bagian output yaitu *GlobalAveragePooling2D*, *Dense(1024, activation= Relu, Dropout(0.5), dan Dense(class_count, activation=softmax))* yang dapat di tinjau pada gambar 5.

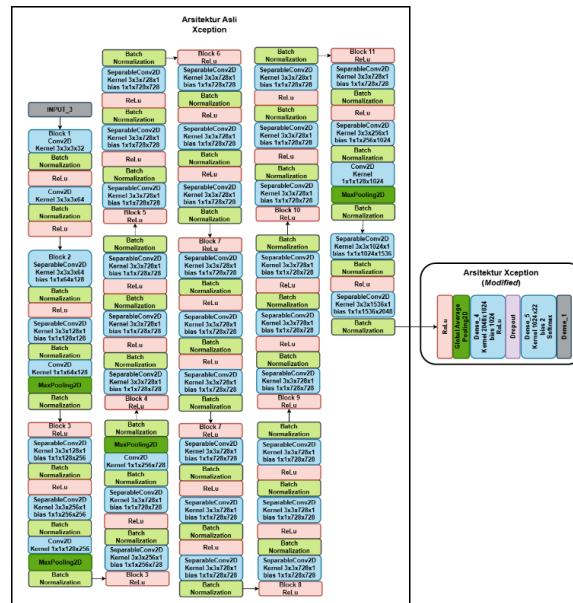


Gambar 5. Arsitektur DenseNet121(Modified)

Sebelum di modifikasi, jumlah total parameter pada model summary sebanyak 7,037,504 dan setelah dilakukan modifikasi jumlahnya naik menjadi 8,089,154. Ini menunjukkan bahwa 4 layer yang di masukan ke layer *output* mempengaruhi meningkatnya jumlah parameter karena layer dense yang digunakan memiliki jumlah *neuron* sebanyak 1024. Dengan jumlah *neuron* yang banyak ini, tentu sangat mempengaruhi total parameter yang di dapatkan setelah melakukan modifikasi. Dengan besarnya parameter ini diharapkan dapat meningkatkan performa pelatihan model sehingga mendapatkan hasil akurasi terbaik dari model arsitektur sebelumnya.

3) *Xception (Modified)*

Xception (Modified) adalah salah satu arsitektur yang telah dimodifikasi dari arsitektur aslinya yaitu *Xception* yang memiliki hampir 71 lapis layers dan menggunakan pendekatan deepwise separable convolutions di hampir setiap lapisan. Pada *Xception (Modified)* dilakukan pula pemotongan layer sebanyak 30 layer bagian akhir dan menambahkan 4 lapisan layer baru di bagian *output* yaitu : GlobalAveragePooling2D, Dense(1024, activation= Relu, Dropout(0.5), dan Dense(class_count, activation=softmax)) yang dapat di tinjau pada tabel I. Hal ini dilakukan untuk meningkatkan hasil pengujian yang dilakukan oleh *Xception (Modified)* sehingga mampu menciptakan model yang berkualitas dengan memperbaharui model sebelumnya. Berikut adalah gambar arsitektur *Xception (Modified)* yang dapat di tinjau pada gambar 6.

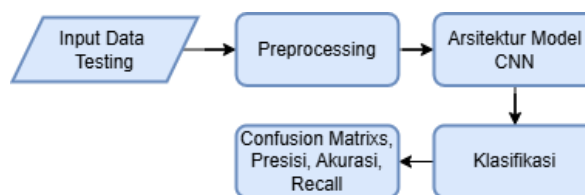


Gambar 6. Arsitektur *Xception(Modified)*

Arsitektur ini sama dengan arsitektur lainnya yang dimodifikasi, jumlah total parameter pada arsitektur *Xception (Modified)* mengalami kenaikan, yang sebelum di modifikasi berjumlah 20,861,480 dan setelah di modifikasi berjumlah 2,2961,706. Dengan ini membuktikan bahwa Ketika layer pada arsitektur yang telah dipotong 30 bagian hanya mengurangi sedikit jumlah parameternya, sedangkan 4 layer yang di tambahkan memiliki jumlah *neuron* pada dense sebanyak 1024 yang mempengaruhi jumlah parameter, sehingga jumlah total parameter mengalami kenaikan.

2.5 Testing

Testing adalah proses dimana dataset yang telah di *Pre-processing* dan telah di lakukan pemodelan akan di hitung dan di proses menggunakan model yang sudah di buat Pradana *et al.* (2022). Dalam penelitian ini, ada beberapa langkah yang di lakukan yaitu, Arsitektur Model yang di gunakan, Klasifikasi, dan Confusion Matrix yang melibatkan Akurasi, Presisi, Recall, dan F1 Score. Alur proses testing di tunjukkan pada gambar 7.



Gambar 7. Alur Tahap *Testing*

Pada tahap *testing* dilakukan yaitu memasukkan data testing yang di proses pada tahap training sebelumnya. Setelah itu, dilakukan proses *Pre-processing* Kembali untuk meningkatkan kualitas data. Langkah selanjutnya melakukan pengujian model dari arsitektur *MobileNetV2*, *MobileNetV2 (Modified)*, *DenseNet121*, *DenseNet121(Modified)*, *Xception*, dan *Xception (Modified)*. Pengujian pada model arsitektur yang tidak dimodifikasi hanya di ubah parameternya saja, sedangkan untuk arsitektur yang dimodifikasi (*Modified*) ada beberapa perubahan, seperti memotong 30 layer terakhir, kemudian menambahkan 4 layer baru yang akan di tunjukkan pada tabel 3. Setelah itu, model akan di simpan ke dalam format H5 dan kemudian akan digunakan untuk melakukan klasifikasi.

2.6 Performance Calculation

Performance Calculation digunakan untuk melihat seberapa akurat atau seberapa besar akurasi dan error yang di dapat melalui model dan proses-proses yang telah di lalui Khoiruddin *et al.* (2022). Cara ini dilakukan untuk mengevaluasi model yang di buat melalui arsitektur orisinal ataupun arsitektur yang telah di modifikasi. Hasil kalkulasi yang dihitung adalah *accuracy*, *precision*, *recall*, dan *f1 score* secara dengan meninjau persamaan (1), (2), (3), dan (4). Dengan keterangan TP yang berarti Positif True, TN yang berarti True Negatif, FP yang berarti False Positif, dan FN yang berarti False Negatif Kusuma *et al.* (2023).

$$Accuracy = \frac{TP + TN}{TP + FP + TNFN}$$

$$Recall = \frac{TP+TN}{TP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$F1\ Score = \frac{Precision * Recall}{Precision + Recall}$$

3. Hasil dan Pembahasan

3.1 Hasil

Berdasarkan hasil penelitian yang telah dilakukan untuk mengklasifikasi Batik Lampung dan Batik Yogyakarta menggunakan metode *Convolutional Neural Network* dengan 6 arsitektur yaitu yang terdiri dari 3 arsitektur yang tidak di modifikasi dan 3 arsitektur yang di modifikasi (*Modified*) yaitu *MobileNetV2*, *DenseNet121*, *Xception*, *MobileNetV2 (Modified)*, *DenseNet121 (Modified)*, dan *Xception (Modified)*. Masing-masing arsitektur di atas telah dilakukan uji coba eksperimen dengan ketentuan parameter yang berbeda-beda seperti *Learning rate* dan epoch serta menampilkan hasil terbaik yang akan di tunjukkan pada tabel di bawah ini.

Tabel 4. Hasil Eksperimen Arsitektur Yang Tidak Dimodifikasi

Arsitektur	Learning rate	Epoch	Akurasi	Presisi	Recall	F1 Score	Loss
<i>MobileNetV2</i>	0,01	10	90,43%	88,81%	84,34%	80,87%	0,3780
<i>MobileNetV2</i>	0,001	20	93,53%	92,77%	87,66%	89,55%	0,2311
<i>MobileNetV2</i>	0,0001	30	96,20%	97,93%	95,77%	96,45%	0,1927
<i>DenseNet121</i>	0,01	10	79,89%	70,22%	78,95%	82,66%	0,5764
<i>DenseNet121</i>	0,001	20	89,88%	84,59%	80,69%	91,71%	0,4657
<i>DenseNet121</i>	0,0001	30	95,06%	92,85%	97,82%	95,65%	0,3081
<i>Xception</i>	0,01	10	89,77%	89,66%	81,56%	84,62%	0,2549
<i>Xception</i>	0,001	20	94,56%	93,20%	90,10%	97,34%	0,1210
<i>Xception</i>	0,0001	30	97,53%	97,22%	96,05%	97,02%	0,0798

Dari penjelasan tabel di atas diketahui bahwa, eksperimen dilakukan pada *learning rate* menggunakan 0,01, 0,001, dan 0,0001 serta epoch 10, 20, dan 30 terhadap 3 arsitektur *MobileNetV2*, *DenseNet121*, dan *Xception*. Dari ketiga arsitektur tersebut yang mendapatkan hasil terbaik adalah *EXception* dengan perolehan akurasi 97,53%, Presisi 97,22%, Recall 96,05%, F1 Score 97,02%, dan Loss 0,0798 pada *learning rate* 0,0001 dan epoch 30 sehingga menjadikan *Xception* model terbaik pada

eksperimen arsitektur yang tidak dimodifikasi. Selanjutnya eksperimen yang dilakukan pada arsitektur yang dimodifikasi (*Modified*) dengan yaitu *MobileNetV2 (Modified)*, *DenseNet121 (Modified)*, dan *Xception (Modified)*. Arsitektur yang dimodifikasi melakukan eksperimen yang sama dengan yang tidak dimodifikasi yaitu dengan mengganti parameter *learning rate* yaitu 0,01, 0,001, dan 0,0001 serta epoch 10, 20, dan 30. Akan tetapi, arsitektur yang dimodifikasi juga mengubah layer dengan memotong 30 layer terakhir dan menambah 4 layer baru di atas base modelnya sehingga itu yang menjadikannya berbeda dengan arsitektur yang tidak dimodifikasi. Hasil dari eksperimen terbaik akan di tunjukkan pada tabel 5 di bawah ini.

Tabel 5. Hasil Eksperimen Arsitektur Yang Dimodifikasi

Arsitektur	Learning rate	Epoch	Akurasi	Presisi	Recall	F1 Score	Loss
<i>MobileNetV2</i>	0,01	10	88,99%	80,33%	72,77%	77,67%	0,1121
<i>MobileNetV2</i>	0,001	20	90,23%	89,70%	79,88%	83,93%	0,0678
<i>MobileNetV2</i>	0,0001	30	98,32%	92,80%	89,90%	90,88%	0,0199
<i>DenseNet121</i>	0,01	10	90,89%	84,80%	86,46%	79,80%	0,1674
<i>DenseNet121</i>	0,001	20	94,56%	89,87%	91,78%	84,80%	0,1342
<i>DenseNet121</i>	0,0001	30	99,09%	94,98%	96,09%	93,45%	0,0876
<i>Xception</i>	0,01	10	94,88%	89,09%	85,80%	80,34%	0,0342
<i>Xception</i>	0,001	20	97,50%	93,57%	89,58%	86,88%	0,0180
<i>Xception</i>	0,0001	30	100%	97,49%	94,97%	92,66%	0,0066

Pada tabel di atas menunjukkan bahwa, arsitektur yang dimodifikasi (*Modified*) menggunakan parameter yang sama dengan arsitektur yang tidak dimodifikasi yang menggunakan *learning rate* 0,01, 0,001, dan 0,0001 serta epoch 10, 20, dan 30. Yang menjadi pembeda adalah arsitektur yang di modifikasi telah diubah layernya dengan memotong 30 layer terakhir dan menambahkan 4 layer baru di atas base modelnya. Dengan perubahan tersebut, hasil yang di dapatkan oleh arsitektur ini cukup mengesankan dengan mendapatkan akurasi terbaik pada model *Xception (Modified)* dengan akurasi sebesar 100%, Presisi 97,49%, Recall 94,97%. F1 Score 92,66%, dan Loss 0,0066 pada eksperimen epoch ke 30 dan *learning rate* 0,0001. Dengan demikian, hasil akurasi terbaik pada kedua model yaitu arsitektur yang tidak dimodifikasi dan arsitektur yang dimodifikasi (*Modified*) akan di tunjukkan pada tabel 6 di bawah ini.

Tabel 6. Hasil Eksperimen Arsitektur Terbaik Learning rate 0,0001 dan Epoch 30

Matriks	<i>Xception</i>	<i>Xception(Modified)</i>
Akurasi	97,53%	100%
Presisi	97,22%	97,49%
Recall	96,05%	94,97%
F1 Score	97,02	92,66%
Loss	0,0798	0,0066

Dari gambar tabel di atas, bahwa arsitektur *Xception (Modified)* lebih unggul dengan akurasi mencapai 100%, Presisi 97,49%, Recall 94,97%. F1 Score 92,66%, dan Loss 0,0066 pada eksperimen epoch ke 30 dan *learning rate* 0,0001 sedangkan arsitektur *Xception* mendapatkan akurasi sebesar 97,53%, Presisi 97,22%, Recall 96,05%, F1 Score 97,02%, dan Loss 0,0798 pada *learning rate* 0,0001 dan epoch 30. Hasil tersebut membuktikan bahwa arsitektur *Xception (Modified)* memiliki layer yang banyak di antara arsitektur lainnya sehingga arsitektur tersebut mendapatkan hasil yang lebih baik dari kedua arsitektur lainnya walaupun sama-sama sudah di modifikasi dengan cara yang sama, tetapi parameter yang berbeda. Selain itu, penelitian ini juga memiliki keterbatasan seperti hanya melakukan training sebanyak 30 epoch sehingga ada beberapa arsitektur yang memiliki loss yang kurang stabil yang mempengaruhi kinerja model saat melakukan klasifikasi. Selanjutnya untuk mengevaluasi hasil kinerja model pada arsitektur yang dimodifikasi (*Modified*) dan tidak dimodifikasi, berikut merupakan

gambar yang akan menjadi evaluasi dari kedua model tersebut. Penjelasan akan di tinjau pada gambar 8 dan Gambar 9.



Gambar 8. Grafik Arsitektur *Xception*



Gambar 9. Grafik Arsitektur *Xception(Modified)*

Pada gambar 8 dan gambar 9 merupakan gambar diagram evaluasi terhadap model arsitektur yang dimodifikasi (*Modified*) dan yang tidak dimodifikasi. Terlihat bahwa *train and validation loss* di kedua grafik tersebut memiliki evaluasi yang cukup mumpuni dimana antara *validation loss* arsitektur yang dimodifikasi dan yang tidak di modifikasi memiliki keseimbangan yang membuat kedua grafik tersebut tidak jauh berbeda. Hanya saja, grafik pada arsitektur yang dimodifikasi memiliki *validation accuracy* yang menyentuh angka 100% yang dimana Model *Xception (Modified)* merupakan model arsitektur terbaik.

3.2 Pembahasan

Batik sebagai warisan budaya Indonesia, memiliki nilai estetika dan makna yang mendalam. Dalam klasifikasi batik, berbagai metode telah diterapkan untuk mengenali dan mengidentifikasi motif batik dengan akurasi tinggi. Penelitian oleh Andono dan Rachmawanto (2021) menunjukkan bahwa penggunaan *Multikernel SVM* dengan ekstraksi fitur *GLCM* dan *LBP* dapat mencapai akurasi 100%, menegaskan pentingnya pemilihan fitur yang tepat dalam meningkatkan hasil klasifikasi. Di sisi lain, Ayu Ratna Juwita *et al.* (2021) menerapkan *Convolutional Neural Network (CNN)* untuk identifikasi citra batik dan mendapatkan hasil yang menjanjikan, sejalan dengan penelitian oleh Masa dan Hamdani (2021) yang juga menggunakan *CNN* berdasarkan *K-means Clustering*, mencapai akurasi yang sama. Hal ini menunjukkan bahwa *CNN* merupakan metode yang efektif dalam klasifikasi citra batik, terutama dalam mengenali pola kompleks. Sementara itu, penelitian oleh Utari dan Zulfikar (2023) menggunakan *CNN* dengan teknik deteksi tepi untuk identifikasi motif batik, yang menunjukkan bahwa inovasi dalam teknik pengolahan citra dapat memberikan hasil yang lebih baik. Agustiani *et al.* (2022) melakukan klasifikasi penyakit pada daun padi menggunakan *Random Forest* dan *Color Histogram*, yang menunjukkan bahwa pendekatan berbasis fitur warna juga dapat diterapkan dalam konteks klasifikasi batik, membuka peluang untuk mengeksplorasi metode lain yang mungkin belum banyak diterapkan.

Meskipun banyak penelitian menunjukkan hasil menjanjikan, tantangan dalam pengumpulan data dan pengujian model tetap ada. Hawibowo dan Muhimmah (2024) menyoroti pentingnya aplikasi berbasis Android untuk mendeteksi tingkat kematangan pepaya menggunakan CNN, yang relevan untuk klasifikasi batik, di mana aplikasi mobile dapat membantu masyarakat mengenali produk dengan lebih baik. Penelitian ini berfokus pada tiga arsitektur: *MobileNetV2*, *DenseNet121*, dan *Xception*, yang dimodifikasi untuk meningkatkan akurasi. Penelitian oleh Putra *et al.* (2023) menunjukkan bahwa penggunaan *Transfer Learning* dalam CNN dapat meningkatkan akurasi dalam klasifikasi jenis rempah-rempah, yang menunjukkan bahwa pendekatan serupa dapat diterapkan dalam klasifikasi batik untuk mendapatkan hasil yang lebih baik dan stabil. Penelitian tentang klasifikasi batik menunjukkan bahwa penggunaan teknologi canggih seperti CNN dan metode ekstraksi fitur lainnya sangat efektif dalam mengenali dan mengidentifikasi motif batik, dan dengan terus melakukan inovasi serta penelitian lebih lanjut, diharapkan pemahaman masyarakat tentang batik dapat meningkat dan mendorong pelestariannya sebagai warisan budaya yang berharga.

4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan untuk mengklasifikasi Batik Lampung dan Batik Yogyakarta menggunakan metode *Convolutional Neural Network* dengan 6 arsitektur yaitu *MobileNetV2*, *DenseNet121*, dan *Xception* yang tidak dimodifikasi dan *MobileNetV2 (Modified)*, *DenseNet121 (Modified)*, dan *Xception (Modified)* yang dimodifikasi menghasilkan perbandingan akurasi yang cukup signifikan. Dimana masing-masing arsitektur memiliki eksperimen yang digunakan dalam pelatihan model dengan mengubah parameter dan memotong layer dalam upaya untuk mencari hasil yang terbaik. Dalam model arsitektur yang tidak dimodifikasi, Arsitektur *Xception* unggul dengan hasil akurasi di angka 97,53%, Presisi 97%, Recall 96%, F1 Score 97%, Loss 0,0798 pada epoch 30 dan *learning rate* 0,0001. Sedangkan arsitektur yang dimodifikasi (*Modified*) mendapatkan hasil yang sangat memuaskan yaitu akurasi sebesar 100%, Presisi 97%, Recall 94%, F1 Score 92%, dan Loss 0,0066 pada epoch 30 dan *learning rate* 0,0001. Hal ini membuktikan bahwa arsitektur yang tidak dimodifikasi sudah memiliki hasil akurasi yang cukup baik, namun masih sangat kurang baik pada hasil loss, sehingga arsitektur yang dimodifikasi ini untuk menyelesaikan masalah sebelumnya sehingga hasil akurasi dan loss seimbang dan menjadi model terbaik dalam penelitian ini. Pada penelitian selanjutnya akan jauh lebih baik untuk menggunakan banyak arsitektur lain untuk menguji seberapa besar pendekatan dari masing-masing arsitektur serta melakukan banyak eksperimen sebagai media untuk mencari nilai solusi terbaik dengan memperbanyak proses epoch untuk menapatkan akurasi yang akurat, lebih banyak melibatkan arsitektur dalam melakukan uji coba model sehingga menjadikan metode *Convolutional Neural Network* sebagai metode paling baik untuk melakukan pendekatan pada setiap citra dengan beragam arsitektur untuk melakukan klasifikasi atau deteksi dalam kasus mempromosikan budaya atau kesenian lainnya dalam upaya menemukan solusi yang terbaik.

5. Daftar Pustaka

- Agustiani, S., Arifin, Y. T., Junaidi, A., Wildah, S. K., & Mustopa, A. (2022). Klasifikasi Penyakit Daun padi Menggunakan random forest dan color histogram. *Jurnal Komputasi*, 10(1), 65-74. <https://doi.org/10.23960/komputasi.v10i1.2961>.
- Andono, P. N., & Rachmawanto, E. H. (2021). Evaluasi Ekstraksi Fitur GLCM dan LBP Menggunakan Multikernel SVM untuk Klasifikasi Batik. *Jurnal Resti (Rekayasa Sistem dan Teknologi Informatika)*, 5(1), 1-9. <https://doi.org/10.29207/resti.v5i1.2615>.

- ATMAJA, W. P., & LUSIANA, V. (2023). Klasifikasi Jenis Batik Pekalongan Menggunakan Citra Hsi Dengan Metode K-Nearest Neighbor. *Jurnal INSTEK (Informatika Sains dan Teknologi)*, 8(1), 1-7. <https://doi.org/10.24252/instek.v8i1.35967>.
- Fahcuroji, A. R., Wijaya, M. Y., & Fauziah, I. (2024). IMPLEMENTASI ALGORITMA CNN MOBILENET UNTUK KLASIFIKASI GAMBAR SAMPAH DI BANK SAMPAH. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, 11(1), 45-51. <https://doi.org/10.30656/prosisko.v11i1.8101>.
- Harahap, R. I. J., & Khairani, S. (2024). Implementasi Metode K-Nearest Neighbor Untuk Klasifikasi Penyakit Tanaman Mentimun Pada Citra Daun. *Jurnal Ilmu Komputer dan Sistem Informasi*, 3(2), 135-146. <https://doi.org/10.70340/jirsi.v3i2.123>.
- Hawibowo, M. S., & Muhimmah, I. (2024). Aplikasi Pendeteksi Tingkat Kematangan Pepaya menggunakan Metode Convolutional Neural Network (CNN) Berbasis Android. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 10(1), 162-170. <https://doi.org/10.26418/jp.v10i1.77819>.
- Juwita, A. R., Al Mudzakir, T., Pratama, A. R., Husodo, P., & Sulaiman, R. (2021). IDENTIFIKASI CITRA BATIK DENGAN METODE CONVOLUTIONAL NEURAL NETWORK. *BUANA ILMU*, 6(1), 192-208.
- Khoiruddin, M., Junaidi, A., & Saputra, W. A. (2022). Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network. *Journal of Dinda: Data Science, Information Technology, and Data Analytics*, 2(1), 37-45. <https://doi.org/10.20895/dinda.v2i1.341>.
- Kusuma, J., Rosnelly, R., & Hayadi, B. H. (2023). Klasifikasi Penyakit Daun Pada Tanaman Jagung Menggunakan Algoritma Support Vector Machine, K-Nearest Neighbors dan Multilayer Perceptron. *Journal of Applied Computer Science and Technology*, 4(1), 1-6. <https://doi.org/10.52158/jacost.v4i1.484>.
- 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), 1292-1301.
- Maulana, I., Sastypratiwi, H., Muhardi, H., Safriadi, N., & Sujaini, H. Implementasi Convolutional Neural Network (CNN) untuk Klasifikasi Motif Batik pada Aplikasi Computer Vision Berbasis Android. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 9(3), 384-393. <https://doi.org/10.26418/jp.v9i3.69496>.
- Nisa, C., & Candra, F. (2024). Klasifikasi Jenis Rempah-Rempah Menggunakan Algoritma Convolutional Neural Network: Classification of Spice Types Using the Convolutional Neural Network Algorithm. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(1), 78-84. <https://doi.org/10.57152/malcom.v4i1.1018>.
- Pradana, D., Alghifari, M. L., Juna, M. F., & Palaguna, D. (2022). Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network. *Indonesian Journal of Data and Science*, 3(2), 55-60. <https://doi.org/10.56705/ijodas.v3i2.35>.

- Putra, A. E., Naufal, M. F., & Prasetyo, V. R. (2023). Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 9(1), 12-18. <https://doi.org/10.26418/jp.v9i1.58186>.
- Raharja, A. D. R. J., & Widyastuti, W. (2022, July). Pengenalan Motif Batik Jogja dengan Jaringan Syaraf Tiruan Perambatan Balik. In *Seminar Nasional Riset & Inovasi Teknologi* (Vol. 1, No. 1, pp. 214-222).
- Utari, L., & Zulfikar, A. (2023). Penerapan Convolutional Neural Networks Menggunakan Edge Detection Untuk Identifikasi Motif Jenis Batik. *TeknoIS: Jurnal Ilmiah Teknologi Informasi dan Sains*, 13(1), 110-123. <https://doi.org/10.36350/jbs.v13i1.184>.
- Yanuardi, M. I., Aminudin, A., & Faiqurahman, M. Analisis Perbandingan Efektivitas Arsitektur Restful dan Arsitektur Grpc pada Implementasi Web Service. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 10(2), 333-341. <https://doi.org/10.26418/jp.v10i2.80845>.