

DAFTAR PUSTAKA

- [1] Ackerman, S., Farchi, E., Raz, O., Zalmanovici, M., & Dube, P. (2020). *Detection of data drift and outliers affecting machine learning model performance over time*. <http://arxiv.org/abs/2012.09258>
- [2] Gomes, J. C., & Borges, D. L. (2022). Insect Pest Image Recognition: A Few-Shot Machine Learning Approach including Maturity Stages Classification. *Agronomy*, 12(8). <https://doi.org/10.3390/agronomy12081733>
- [3] Herulambang, W., Hamidah, M. N., & Setyatama, F. (2020, February 1). Comparison of SVM and BPNN Methods in the Classification of Batik Patterns Based on Color Histograms and Invariant Moments. *Proceeding - ICoSTA 2020: 2020 International Conference on Smart Technology and Applications: Empowering Industrial IoT by Implementing Green Technology for Sustainable Development*. <https://doi.org/10.1109/ICoSTA48221.2020.1570615583>
- [4] Hino, H. (2020). *Active Learning: Problem Settings and Recent Developments*. <http://arxiv.org/abs/2012.04225>
- [5] Hor, S. L., AlDahoul, N., Karim, H. A., Lye, M. H., Mansor, S., Fauzi, M. F. A., & Ba Wazir, A. S. (2022). Deep Active Learning for Pornography Recognition Using ResNet. *International Journal of Technology*, 13(6), 1261–1270. <https://doi.org/10.14716/ijtech.v13i6.5842>
- [6] Hou, M., & Sato, I. (2021). *A Closer Look at Prototype Classifier for Few-shot Image Classification*. <http://arxiv.org/abs/2110.05076>
- [7] Ibda, H. (2021). STRATEGI MEMUTUS MATA RANTAI PEMBAJAKAN HAK CIPTA PADA SENI BATIK NUSANTARA. In *Citra Ilmu, Edisi* (Vol. 33). <https://kumparan.com/haikalkurniawan/pentingnya->
- [8] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). *A ConvNet for the 2020s*. <http://arxiv.org/abs/2201.03545>

- [9] Martinez, I., Viles, E., & Olaizola, I. G. (2022). *A survey study of success factors in data science projects*. <https://doi.org/10.1109/BigData52589.2021.9671588>
- [10] Parnami, A., & Lee, M. (2022). *Learning from Few Examples: A Summary of Approaches to Few-Shot Learning*. <http://arxiv.org/abs/2203.04291>
- [11] Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., & Dollár, P. (2020). *Designing Network Design Spaces*. <http://arxiv.org/abs/2003.13678>
- [12] Rasyidi, M. A., Handayani, R., & Aziz, F. (2021). Identification of batik making method from images using convolutional neural network with limited amount of data. *Bulletin of Electrical Engineering and Informatics*, 10(3), 1300–1307. <https://doi.org/10.11591/eei.v10i3.3035>
- [13] Shi, F., Wang, Z., Hu, M., & Zhai, G. (2020). Active learning plus deep learning can establish cost-effective and robust model for multichannel image: A case on hyperspectral image classification. In *Sensors (Switzerland)* (Vol. 20, Issue 17, pp. 1–14). MDPI AG. <https://doi.org/10.3390/s20174975>
- [14] Song, G., Tao, Z., Huang, X., Cao, G., Liu, W., & Yang, L. (2020). Hybrid Attention-Based Prototypical Network for Unfamiliar Restaurant Food Image Few-Shot Recognition. *IEEE Access*, 8, 14893–14900. <https://doi.org/10.1109/ACCESS.2020.2964836>
- [15] Tan, M., & Le, Q. V. (2021). *EfficientNetV2: Smaller Models and Faster Training*. <http://arxiv.org/abs/2104.00298>
- [16] Zhang, B., Gao, B., Liang, S., Li, X., & Wang, H. (2023). A classification algorithm based on improved meta learning and transfer learning for few-shot medical images. *IET Image Processing*. <https://doi.org/10.1049/ipr2.12889>

Turnitin_Jurnal_Adil Faruq

Habibi_Informatika.Turnitin_Jurnal_Adil Faruq

Habibi_Informatika

ORIGINALITY REPORT

4%

SIMILARITY INDEX

4%

INTERNET SOURCES

2%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

1

pdfs.semanticscholar.org

Internet Source

1%

2

www.researchgate.net

Internet Source

1%

3

travel.kompas.com

Internet Source

<1%

4

jurnal.unsil.ac.id

Internet Source

<1%

5

es.scribd.com

Internet Source

<1%

6

nero.trunojoyo.ac.id

Internet Source

<1%

7

ejournal.raharja.ac.id

Internet Source

<1%

8

cogito.unklab.ac.id

Internet Source

<1%

ejournal.stmik-budidarma.ac.id

Skripsi_Adil Faruq Habibi_Informatika

ORIGINALITY REPORT

11%

SIMILARITY INDEX

10%

INTERNET SOURCES


3%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES



1	Submitted to Southville International School and Colleges Student Paper	2%
2	etheses.uin-malang.ac.id Internet Source	1%
3	Submitted to CSU, San Jose State University Student Paper	1%
4	etd.repository.ugm.ac.id Internet Source	<1%
5	library.palcomtech.com Internet Source	<1%
6	docplayer.info Internet Source	<1%
7	repository.usbypkp.ac.id Internet Source	<1%
8	repository.ub.ac.id Internet Source	<1%
9	Submitted to Universitas Palangka Raya Student Paper	<1%



Optimasi Pengenalan dan Pelabelan Motif Batik dengan Prototypical Networks dan Uncertainty Analysis

Adil Faruq Habibi¹, Fauziah², Aris Gunaryati³

¹ Fakultas Teknologi Komunikasi dan Informatika, Informatika, Universitas Nasional, Jakarta, Indonesia

² Fakultas Teknologi Komunikasi dan Informatika, Magister Teknologi Informasi, Universitas Nasional, Jakarta, Indonesia

³ Fakultas Teknologi Komunikasi dan Informatika, Sistem Informasi, Universitas Nasional, Jakarta, Indonesia

Email: ¹adilfaruqhabibi@gmail.com, ²fauziah@civitas.unas.ac.id, ³aris.gunaryati@civitas.unas.ac.id

Email Penulis Korespondensi: adilfaruqhabibi@gmail.com

Abstrak— Indonesia dikenal dengan kekayaan budaya batiknya yang mencakup ribuan motif unik dari berbagai daerah. Mengidentifikasi dan melabeli motif-motif ini secara manual adalah tugas yang kompleks dan memakan waktu. Meskipun telah ada penelitian yang menggunakan machine learning untuk pengenalan motif batik, pendekatan tersebut seringkali terbatas pada jumlah kelas yang tetap. Jika kelas baru ditemukan, biasanya dibutuhkan banyak data tambahan, dan model harus dilatih ulang, yang menjadi tantangan tersendiri. Penelitian ini bertujuan untuk mengoptimalkan proses pengenalan dan pelabelan motif batik dengan menggabungkan Prototypical Networks (ProtoNets) dengan Uncertainty Analysis. Metode ini menawarkan fleksibilitas dalam pengenalan motif, memungkinkan penambahan kelas tanpa perlu melatih ulang model. Dari hasil eksperimen ini saat menggunakan pretrained ConvNeXt Tiny sebagai backbone ProtoNets memperoleh akurasi hingga 92.6% dalam pengenalan kelas-kelas novel. Selain itu saat dihadapkan pada tugas pelabelan terhadap 120 data dari kelas-kelas novel, pendekatan ini dapat menghemat biaya pelabelan hingga 76%. Platform web yang dikembangkan juga menyediakan fitur pengenalan motif batik, memberikan analitik melalui uncertainty analysis, dan visualisasi Class Activation Map yang membantu dalam interpretasi model. Dan juga terdapat fitur pelabelan interaksi model dan manusia. Penelitian ini menawarkan solusi efisien dan fleksibel untuk pengenalan dan pelabelan motif batik, dengan potensi dalam skala yang lebih luas.

Kata Kunci: Motif Batik, Machine Learning; Prototypical Networks; Uncertainty Analysis; ConvNeXt Tiny; Class Activation Map

Abstract— Indonesia is known for its rich batik culture, encompassing thousands of unique motifs from various regions. Identifying and labeling these motifs manually is a complex and time-consuming task. Although there have been studies using machine learning for the recognition of batik motifs, these approaches are often limited to a fixed number of classes. If new classes are discovered, it usually requires a substantial amount of additional data, and the model must be retrained, posing its own challenges. This research aims to optimize the process of recognizing and labeling batik motifs by combining Prototypical Networks (ProtoNets) with Uncertainty Analysis. This method offers flexibility in motif recognition, allowing the addition of classes without the need to retrain the model. From the results of this experiment, when using pretrained ConvNeXt Tiny as the ProtoNets' backbone, an accuracy of up to 92.6% was achieved in recognizing novel classes. Furthermore, when faced with the task of labeling 120 data from novel classes, this approach can save up to 76% of labeling costs. The developed web platform also provides features for recognizing batik motifs, offering analytics through uncertainty analysis, and visualization of the class activation map that assists in model interpretation. There is also a feature for labeling through human-model interaction. This research offers an efficient and flexible solution for recognizing and labeling batik motifs, with potential on a broader scale.

Keywords: Motif Batik, Machine Learning; Prototypical Networks; Uncertainty Analysis; ConvNeXt Tiny; Class Activation Map

1. PENDAHULUAN

Batik merupakan salah satu warisan budaya Indonesia yang memiliki kekayaan ragam jenis yang tidak terhitung jumlahnya. Berdasarkan laporan dari Bandung Fe Institute dan Sobat Budaya pada tahun 2015, ada 5.849 motif batik Indonesia yang tersebar dari Aceh hingga Papua [1]. Saat ini, pengenalan motif batik masih sangat bergantung pada identifikasi manual oleh ahli batik atau pengamat batik yang memiliki pengetahuan mendalam mengenai ragam batik. Namun, dengan perkembangan teknologi machine learning dan deep learning [2], kemampuannya dapat membantu pengenalan motif batik semakin mudah dilakukan bahkan oleh orang awam sekalipun. Dewasa ini, pengembangan sistem machine learning yang reliabel merupakan tugas yang kompleks dan tidak mudah dilakukan. Laporan dari VentureBeat, sebuah media daring yang berbasis teknologi di Amerika Serikat mengatakan bahwa 87% proyek data science tidak pernah diproduksi. Beberapa kegagalan dikarenakan proyek tidak tepat sasaran terhadap penyelesaian masalah dan sebagian besar lainnya dikarenakan *machine learning workflow* dan *development lifecycle* yang kurang efisien dan tidak teroptimasi [3]. Mengetahui bahwa jumlah motif batik yang sangat banyak, diperlukan pendekatan yang fleksibel tidak hanya mampu mengenali batik tapi juga dapat melabeli batik, mengingat dari ribuan motif batik yang ada, ketersediaan dataset yang ada hanya sedikit jumlahnya.

Pada penelitian sebelumnya yang dilakukan oleh Rasyidi et al. berhasil mengklasifikasikan gambar batik berdasarkan cara pembuatannya dengan memperoleh akurasi hingga 87,61% menggunakan pendekatan deep learning dari pretrained Convolutional Neural Network (CNN) [4]. Arsa et al. mengklasifikasikan batik berdasarkan bentuk geometris dan non geometris dengan dengan menggabungkan model Visual Geometry Group 16 (VGG16) sebagai ekstraksi fitur dan Random Forest sebagai klasifikator. Pendekatan ini memperoleh akurasi



hingga 97% [5]. Berbeda dengan dua penelitian sebelumnya, Herulambang et al. mengklasifikasikan 10 motif batik berdasarkan daerah asal dengan akurasi yang diperoleh menggunakan pendekatan machine learning Support Vector Machine (SVM) adalah sebesar 88.33%. Dari ketiga penelitian tersebut dapat disimpulkan bahwa gambar batik dapat diklasifikasikan dengan machine learning atau deep learning. Namun, ketiga penelitian tersebut memiliki keterbatasan terhadap fleksibilitas penambahan jumlah kelas. Pendekatan yang digunakan oleh ketiga penelitian tersebut harus menetapkan sejumlah kelas yang tetap. Sehingga, ketika ada penambahan kelas baru, model perlu dilatih ulang dengan konfigurasi hyperparameter yang baru. Dan perlu diketahui bahwa untuk melatih model machine learning atau deep learning diperlukan jumlah data yang banyak. Oleh karena itu, penelitian ini perlu dikembangkan lagi, mengingat jumlah motif batik yang ada di Indonesia mencapai ribuan.

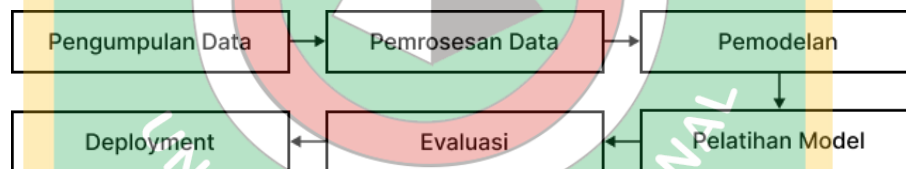
Pada domain lain, terdapat penelitian yang menggunakan pendekatan supervised yang lebih fleksibel terhadap penambahan jumlah kelas ataupun pengenalan terhadap kelas baru. Penelitian yang dilakukan oleh He et al. menyelesaikan permasalahan yang ada di industri nyata terkait diagnosa kerusakan bantalan pada rotari mesin pabrik dengan menggunakan salah satu metode Few-Shot Learning (FSL) yakni, Prototypical Networks (ProtoNets). Pendekatan ini sangat efektif untuk mengatasi keterbatasan data yang susah diperoleh. Hasil dari metode ini mengungguli pendekatan CNN konvensional, yakni dengan hanya menggunakan 5 sampel gambar dapat memperoleh akurasi hingga 90% [6]. Tidak hanya fleksibel untuk pengenalan, pendekatan ini juga dapat digunakan untuk pelabelan. Namun penting untuk dicatat bahwa model machine learning tidak selalu akurat 100%, dan ditambah lagi adanya kemungkinan terdapat data drift di kemudian hari yang dapat menurunkan performa dari model [7]. Oleh karena itu, tugas pelabelan tidak dapat diserahkan sepenuhnya kepada model, tetapi tetap perlu intervensi oleh manusia. Penelitian yang dilakukan oleh Shi et al. mencoba menyelesaikan permasalahan anotasi data gambar hyperspectral blueberry yang mahal dan memakan waktu. Model deep learning yang telah dilatih digunakan untuk pelabelan, dengan bantuan salah satu persamaan Uncertainty Sampling, yakni Entropy Sampling meminimalkan pelabelan secara manual hingga 41.5% [8].

Berdasarkan penelitian sebelumnya menggunakan pendekatan Prototypical Networks dan Uncertainty Sampling sangat fleksibel untuk mengatasi permasalahan pengenalan dan pelabelan motif batik. Oleh karena itu, maka judul dari penelitian ini adalah “Optimasi Pengenalan dan Pelabelan Motif Batik dengan Prototypical Networks dan Uncertainty Analysis”.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian yang digunakan dalam penelitian ini secara keseluruhan ditunjukkan pada Gambar 1.

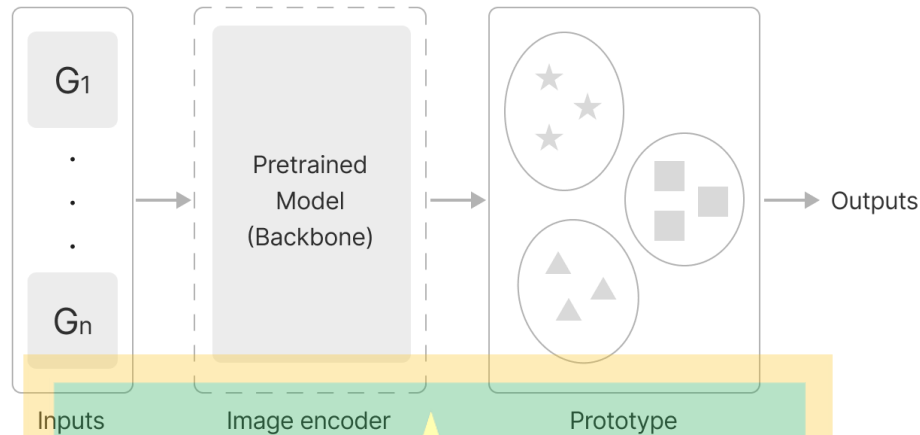


Gambar 1. Tahapan Penelitian

Penelitian ini dimulai dari pengumpulan data, pemrosesan data, pemodelan, pelatihan model, evaluasi dan deployment.

2.1 Prototypical Networks

Pada tahap ini penulis membuat model Prototypical Networks (ProtoNets). ProtoNets merupakan salah satu metode dalam Few-Shot Learning (FSL) yang beroperasi dalam paradigma meta learning [9]. Dalam konteks pengenalan pola atau klasifikasi, FSL bertujuan untuk melatih model yang mampu mengenali objek atau kelas baru dengan hanya menggunakan sedikit contoh atau sampel. ProtoNets mengambil pendekatan berbasis metrik, dimana representasi dari setiap kelas dihitung sebagai prototipe atau titik pusat dari contoh-contoh dalam kelas tersebut. Prototipe ini kemudian digunakan untuk mengklasifikasikan contoh baru berdasarkan jarak metrik terhadap prototipe-prototipe kelas yang ada. Dalam paradigma meta learning, model dilatih untuk ‘belajar bagaimana belajar’, sehingga dapat dengan cepat beradaptasi dengan tugas-tugas baru dengan sedikit data. Pendekatan ini memanfaatkan transfer learning dan struktur hierarkis dalam dataset untuk melakukan generalisasi ke kelas baru dengan sedikit data saja [10]. Berikut adalah arsitektur dari ProtoNets:



Gambar 2. Ilustrasi Arsitektur ProtoNets

Pada Gambar 2 terdapat blok input berupa kumpulan support set (gambar dan label) serta query set (gambar). Lalu di teruskan kepada model pretrained yang menghasilkan vektor embedding untuk masing-masing gambar. Khusus gambar dari support set, dihitung rata-rata embeding-nya yang akan menjadi prototipe. Selanjutnya dihitung jarak embedding gambar dari query set terhadap masing-masing prototipe dengan menggunakan persamaan Euclidean [11].

$$d(p, q) = \sqrt{\sum_{i=1}^n (q - p_i)^2} \tag{1}$$

Pada persamaan 1 diatas, p adalah prototipe masing-masing kelas, n adalah jumlah kelas, dan q adalah vektor embedding dari gambar query. Sehingga dari persamaan tersebut, jarak yang terdekat terhadap salah satu prototipe akan diklasifikasikan sebagai kelas dari label prototipe tersebut. Berikut adalah ringkasan dari cara kerja arsitektur ProtoNets:

- a. Ekstraksi Fitur
Gambar batik diolah untuk diekstrak fitur-fitur penting menjadi vektor embedding yang merepresentasikan motif batik oleh model pretrained sebagai encoder/backbone [12].
- b. Pembentukan Prototipe
Fitur-fitur dari support set kemudian digunakan untuk membentuk prototipe dengan cara mendapatkan nilai rata-rata vektor embedding untuk setiap kelas motif batik.
- c. Pengukuran Jarak
Jarak antara fitur gambar baru atau vektor embedding gambar query dengan prototipe diukur menggunakan persamaan euclidean diatas untuk menentukan kelas motif batik yang sesuai.
- d. Klasifikasi
Berdasarkan jarak yang diukur, gambar query diklasifikasikan ke dalam kelas motif batik yang sesuai.

2.5 Uncertainty Analysis

Uncertainty Sampling adalah teknik active learning di mana model dipilih untuk mengajukan pertanyaan tentang contoh-contoh yang paling tidak yakin jawabannya [13]. Dengan kata lain, model mencoba untuk mengidentifikasi contoh-contoh di mana ia memiliki tingkat kepercayaan terendah dalam prediksinya. Teknik ini dapat digunakan sebagai analisis untuk melihat seberapa bagus model yang dilatih saat melakukan inference terhadap data yang tidak berlabel. Teknik ini juga sering digunakan dalam konteks dimana pelabelan data adalah tugas yang mahal atau memakan waktu, dan tujuannya adalah untuk memaksimalkan efisiensi proses pelabelan dengan memprioritaskan contoh-contoh yang paling mungkin memberikan informasi baru kepada model alias mendeteksi data drift. Berikut adalah beberapa metode uncertainty sampling yang digunakan.

2.5.1 Least Confidence Sampling

$$LC(x) = (1 - P_{\theta}(y^*|x)) \times \frac{n}{n-1} \tag{2}$$

Merupakan selisih antara confidence score tertinggi dengan confidence 100% [14]. Contohnya, jika confidence score tertinggi dari model menyatakan kelas “Batik Bali” ada di dalam gambar, maka least confidence menangkap seberapa yakin (tidak yakin) prediksi tersebut.

2.5.2 Margin of Confidence Sampling

$$MC(x) = 1 - (P_{\theta}(y_1^*|x) - P_{\theta}(y_2^*|x)) \tag{3}$$



Merupakan selisih antara dua prediksi teratas berdasarkan confidence score-nya [15]. Sebagai contoh, jika model paling yakin bahwa “Batik Bali” ada di dalam gambar dan yang kedua paling yakin itu adalah “Batik Dayak”, maka margin of confidence menangkap perbedaan antara dua kepercayaan tersebut.

2.5.3 Ratio of Confidence Sampling

$$RC(x) = \frac{P_{\theta}(y_1^*|x)}{P_{\theta}(y_2^*|x)} \quad (4)$$

Merupakan rasio antara dua prediksi teratas yang paling confidence score-nya. Sebagai contoh, jika model paling yakin bahwa “Batik Bali” ada di dalam gambar dan yang kedua paling yakin adalah itu adalah “Batik Dayak”, maka ratio of confidence menangkap rasio (bukan selisih) antara dua keyakinan.

2.5.4 Entropy-based Sampling

$$ENT(x) = \frac{-\sum_y P_{\theta}(y|x) \log_2 P_{\theta}(y|x)}{\log_2(n)} \quad (5)$$

Merupakan selisih antara semua prediksi dengan mengukur sejauh mana distribusi probabilitas kelas untuk contoh tertentu tersebar. Semakin tinggi entropy, semakin tidak yakin model tentang prediksinya [15].

3. HASIL DAN PEMBAHASAN

Pada bagian ini merupakan bagian krusial dalam penelitian ini, di mana semua temuan eksperimental dan analisis disajikan. Tujuannya adalah menjawab pertanyaan penelitian dan menghubungkan hasil eksperimental dengan teori dan penelitian terdahulu. Selain itu, bab ini juga menguraikan bagaimana penelitian ini berkontribusi pada konteks yang lebih luas.

3.1 Pengumpulan Data

Penulis mengumpulkan data motif batik dari website Kaggle, yang berisi gambar-gambar batik yang telah dikelompokkan berdasarkan asal daerahnya dalam 15 kelas.

3.2 Pemrosesan Data

Tahap ini melibatkan pembuatan dataloader dengan PyTorch, sesuai dengan konfigurasi meta learning. Proses ini meliputi:

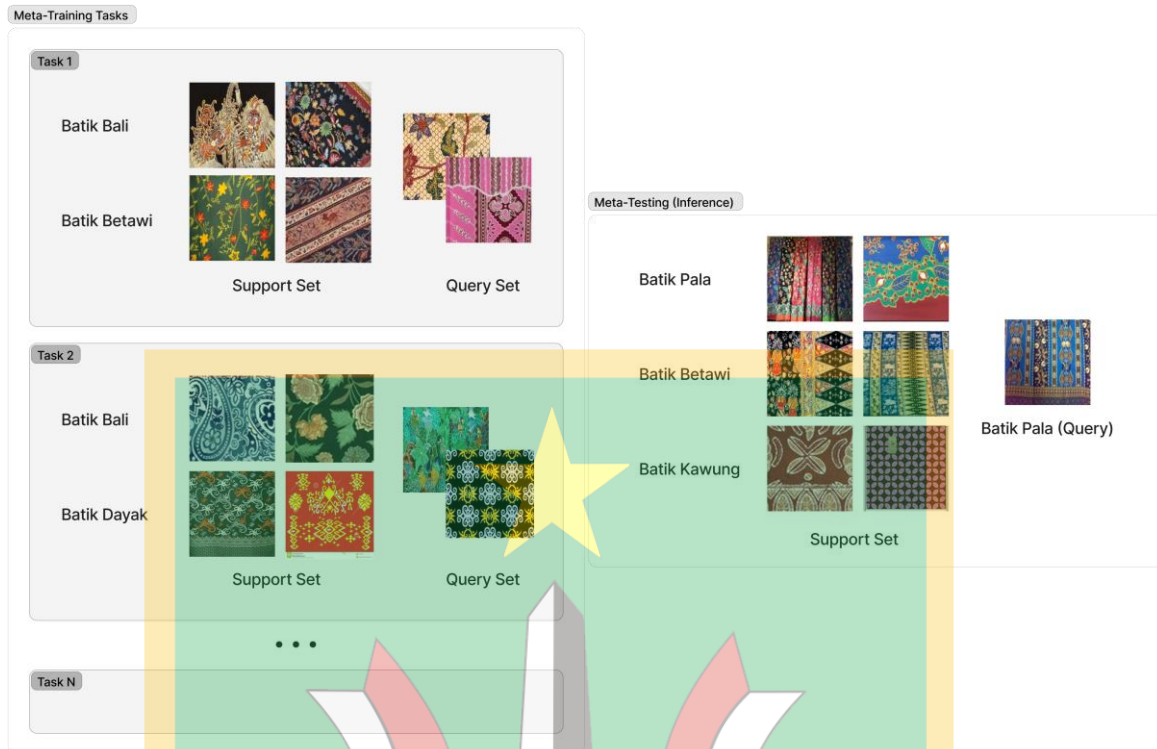
3.2.1 Pemisahan Kelas

Berbeda dengan pendekatan supervised umum yang membagi data latih dan validasi/uji berdasarkan baris, untuk pembagian data pada penelitian ini disesuaikan dengan konfigurasi meta learning dataset. Data dibagi berdasarkan kelas. Sehingga dari 15 kelas yang ada penulis menggunakan 9 kelas untuk data latih dan 6 kelas untuk data validasi/uji. Pemilihan kelas-kelas yang termasuk data latih ataupun data validasi/uji dilakukan secara acak dengan mengatur *random seed* 27 untuk reproduktifitas. Kelas yang tidak pernah dilatih oleh model juga bisa disebut kelas novel. Berikut adalah pembagian kelas data latih dan uji.

- Kelas latih: Batik Pala, Batik Kawung, Batik Parang, Batik Geblek Renteng, Batik Poleng, Batik Dayak, Batik Betawi, Batik Lasem, Batik Cendrawasih.
- Kelas validasi/uji: Batik Tambal, Batik Megamendung, Batik Insang, Batik Bali, Batik Ikat Celup, Batik Sekar Jagad

3.2.2 Batch Sampling

Ini adalah proses pengambilan sekelompok (*batch*) kombinasi data yang akan digunakan dalam satu iterasi pelatihan. Berikut adalah contoh batch sampler sesuai dengan konfigurasi meta learning:



Gambar 3. Contoh Konfigurasi Data Batch Sampler

Pada Gambar 3 terdapat mode training dan mode testing. Pada mode training, terdapat sejumlah task dengan beberapa kelas didalamnya serta terdapat pasangan support set dan query set. Support set adalah kumpulan gambar sesuai dengan kelas yang sudah dilabeli, sedangkan query set adalah beberapa contoh gambar dari tiap kelas yang ada pada support set. Fungsi dari query set adalah untuk mengevaluasi kinerja model dalam mengenali objek yang belum pernah dikenali sebelumnya. Pada mode testing atau inference, kelas-kelas pada support set dapat menggunakan kelas apa pun yang belum pernah dilatih oleh model. Begitu juga dengan query set, yang dapat berupa input gambar yang terdapat dalam support set maupun tidak. Untuk lebih detail, berikut adalah konfigurasi parameter dataloader selama pelatihan dan validasi/pengujian:

Tabel 1. Tabel konfigurasi batch sampler

Parameter	Nilai
n way	3
n shot	10
n query	10
n train task	1500
n val task	200

Parameter beserta nilai yang ada pada Tabel 1 merupakan pengaturan untuk batch sampler dataloader seperti pada Gambar 3. Dalam satu batch terdapat 3 kelas (3-way) dengan masing-masing kelas terdapat 10 gambar (10-shot) dan terdapat 10 gambar query. Jumlah keseluruhan batch pada data latih adalah sebanyak 1500 batch (1500 train task). Sedangkan untuk data validasi/uji terdapat 200 batch (200 val task).

3.2.3 Data Pre-processing

Selama proses pembuatan dataloader, semua gambar satu persatu dilakukan transformasi data. Berikut adalah proses transformasi data latih dan data validasi/uji. Hanya saja untuk data validasi/uji tidak dilakukan augmentasi.

- Resize**
Mengubah ukuran gambar menjadi dimensi (224 x 224) piksel.
- Random Horizontal Flip**
Melakukan pembalikan horizontal secara acak pada gambar. Ini adalah teknik augmentasi data yang membantu model menjadi lebih tangguh terhadap variasi dalam data.
- ToTensor**
Mengubah gambar dari bentuk array NumPy menjadi tensor PyTorch.
- Normalize**
Menormalkan tensor gambar dengan mean dan standard deviasi dari dataset ImageNet. Normalisasi membantu dalam pelatihan yang lebih cepat dan stabil.



3.3 Pelatihan Model

Pelatihan model ProtoNet menggunakan beberapa model pretrained sebagai encoder. Beberapa model pretrained yang digunakan adalah RegNetY_400MF [16], EfficientNetV2-S [17], dan ConvNeXt Tiny [18]. Berikut adalah komparasi tiap-tiap backbone encoder terhadap ProtoNets.

Tabel 2. Tabel perbandingan backbone encoder

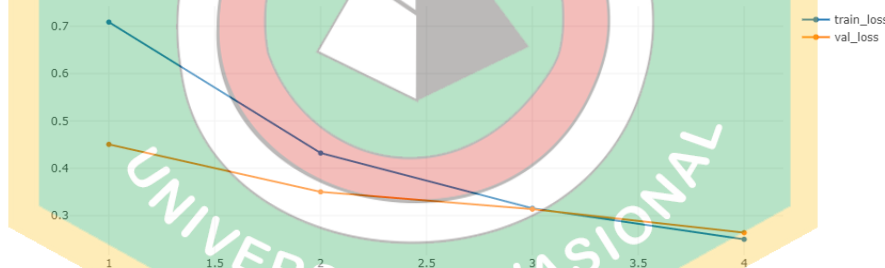
Backbone encoder	Akurasi (%)	Inferensi waktu detik/200 task
RegNetY_400MF	84.7	63.55
EfficientNetV2-S	85.7	103.74
ConvNeXt Tiny	90	55.26

Pada Tabel 2 perbandingan diatas, dapat disimpulkan bahwa encoder/backbone terbaik untuk ProtoNets adalah model pretrained ConvNeXt Tiny. Kesimpulan ini dilihat dari metrik akurasi dan waktu. Akurasi validasi/test model ini tertinggi diantara yang lain sebesar 90%. Selanjutnya waktu inference untuk 200 tugas validasi atau satu epoch dari model ini adalah yang paling cepat dengan waktu 55.26 detik. Untuk evaluasi lebih lanjut berikut hyperparameter yang digunakan ProtoNets (ConvNeXt Tiny) selama pelatihan:

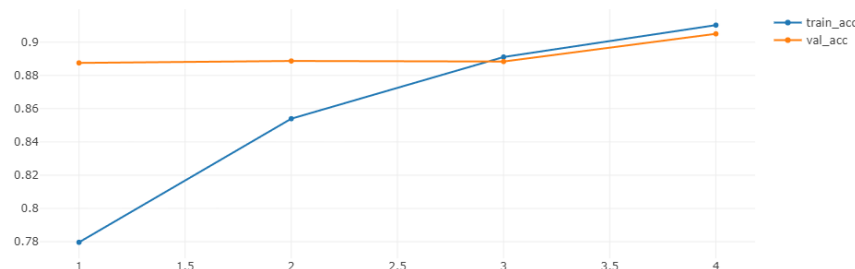
Tabel 3. Tabel hyperparameter yang digunakan

Hyperparameter	Nilai
Dropout	0.2
Epochs	4
Optimizer	SGD
Leaning rate	8e-05
Weight decay	9e-07

Tabel 3 diatas memperlihatkan bahwa model ProtoNets (ConvNeXt Tiny) selama pelatihan menggunakan hyperparameter dropout untuk mengurangi overfitting. Model ini juga dilatih dengan sebanyak 4 epochs. Berikut adalah visualisasi berupa plot garis bagaimana perilaku model selama pelatihan.



Gambar 4. Grafik Training Loss dan Validation Loss



Gambar 5. Grafik Training Akurasi dan Validation Akurasi



Pada Gambar 4 dan Gambar 5 grafik-graik ini memberikan gambaran visual tentang bagaimana model berperilaku selama pelatihan dan dapat membantu dalam mengidentifikasi masalah seperti overfitting atau underfitting. Dan dari grafik ini, dapat disimpulkan bahwa saat pelatihan usai, model ini bisa dikatakan good fit.

3.4 Evaluasi Lanjutan

3.4.1 Perbandingan Skenario Inference

Evaluasi juga dilakukan dengan membandingkan performa model dalam berbagai skenario inference terhadap kelas novel dengan perubahan komposisi berapa banyak jumlah kelas (n way) dan gambar masing-masing kelasnya (n shot). Metrik yang digunakan adalah akurasi, recall, dan precision.

Tabel 4. Tabel perbandingan skenario inference 3-shot

N-way	Akurasi	Precision	Recall
3-way	0.8610	0.8610	0.8791
6-way	0.7223	0.7223	0.7501

Tabel 5. Tabel perbandingan skenario inference 10-shot

N-way	Akurasi	Precision	Recall
3-way	0.9263	0.9263	0.9338
6-way	0.8045	0.8045	0.8166

Tabel 6. Tabel perbandingan skenario inference 20-shot

N-way	Akurasi	Precision	Recall
3-way	0.9110	0.9110	0.9205
6-way	0.8282	0.8282	0.8395

Analisis dari Tabel 4, Tabel 5, dan Tabel 6 membantu penulis dalam memahami bagaimana jumlah kelas dan sampel mempengaruhi hasil. Misalnya, meningkatkan n-way mungkin meningkatkan kompleksitas task, yang dapat mempengaruhi metrik. Demikian pula, perubahan dalam n-shot mungkin memiliki dampak pada seberapa baik model dapat menggeneralisasi dari sedikit contoh [19]. Dan skenario terbaik dari ketiga tabel diatas adalah 3-way dan 10 shot dengan akurasi, precision dan recall berturut-turut adalah 92.6%, 92.6% dan 93%.

3.4.2 Evaluasi Semi-Otomatis Labeling

Dalam penelitian ini, penulis juga melakukan uji coba pelabelan semi-otomatis dengan menggunakan seluruh perhitungan uncertainty sampling yakni, least confidence, margin of confidence, ratio of confidence, dan entropy based. Nilai uncertainty score yang melebihi batas threshold (0.55) ditandai sebagai gambar yang perlu dianotasi oleh manusia. Terdapat 6 novel kelas yang belum pernah dilatih terhadap model sebelumnya. Setiap kelas diuji dengan konfigurasi 3-way (3 kelas support) dan 10-shot (10 gambar pendukung untuk setiap kelas). Jumlah gambar query yang digunakan adalah 20 untuk masing-masing kelas, sehingga total gambar query yang diuji adalah 120. Berikut adalah tabel perbandingan pelabelan secara semi-otomatis:

Tabel 7. Tabel perbandingan pelabelan secara semi-otomatis

Motif Batik	Support Kelas	Prediksi Benar	Perlu Anotasi
Bali	Bali, Tambal, Insang	17/20	7/20
Tambal	Bali, Tambal, Insang	20/20	2/20
Insang	Bali, Tambal, Insang	18/20	5/20
Ikat Celup	Ikat Celup, Megamendung, Sekar Jagad	19/20	5/20
Megamendung	Ikat Celup, Megamendung, Sekar Jagad	18/20	4/20
Sekar Jagad	Ikat Celup, Megamendung, Sekar Jagad	18/20	6/20
Jumlah		110/120	29/120
Persentase		91%	24%

Dari Tabel 7 diatas menunjukkan bahwa model berhasil mengklasifikasikan gambar dengan akurasi prediksi sebesar 91%. Dari total 120 gambar query, terdapat 24% gambar yang memerlukan interaksi manusia dalam proses



pelabelan. Hal ini menunjukkan bahwa model mampu melakukan pelabelan dengan tingkat kepercayaan yang tinggi dan menghemat pelabelan secara manual hingga 76%.

3.5 Deployment

Penulis berhasil mengembangkan sebuah web app yang dapat diakses oleh pengguna lain untuk pengenalan dan pelabelan motif batik. Aplikasi ini dibangun menggunakan Streamlit, sebuah pustaka open-source yang memungkinkan pengembang dengan cepat membuat aplikasi web interaktif untuk ilmu data dan machine learning. Selanjutnya, aplikasi ini diluncurkan di Huggingface, sebuah platform yang menyediakan infrastruktur untuk hosting model-model machine learning. Aplikasi web tersebut dapat diakses melalui tautan berikut, <https://huggingface.co/spaces/afhabibiee/fsl-batik-app>.

3.5.1 Tampilan Antarmuka Pengguna

Aplikasi web yang dikembangkan terdiri dari beberapa bagian utama seperti, halaman home, halaman recognition, dan halaman labeling.



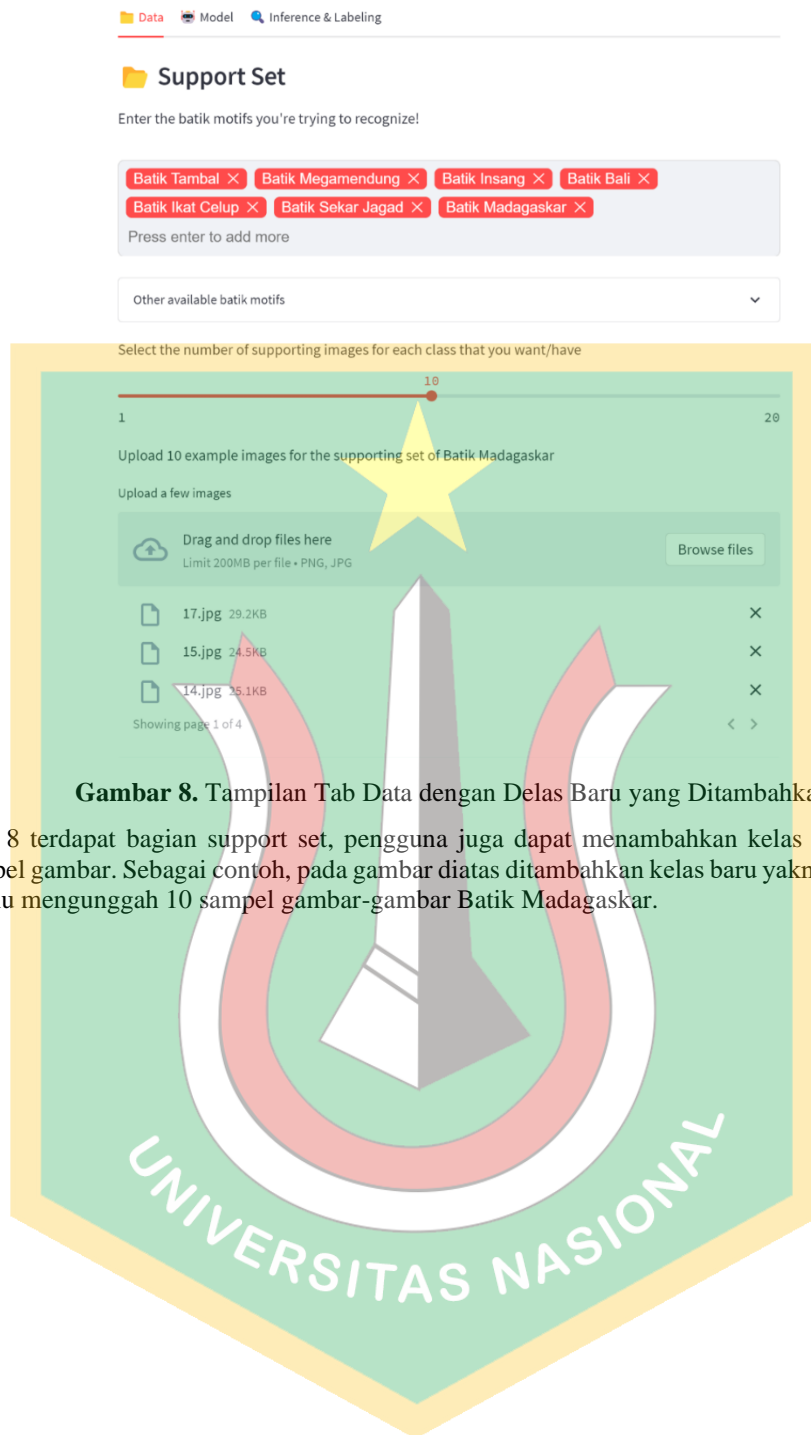
Gambar 6. Tampilan Halaman Home

Pada Gambar 6 terdapat halaman home ini merupakan halaman utama yang berisi peninjauan luas terkait dengan proyek dan terdapat hasil perbandingan untuk berbagai kondisi.



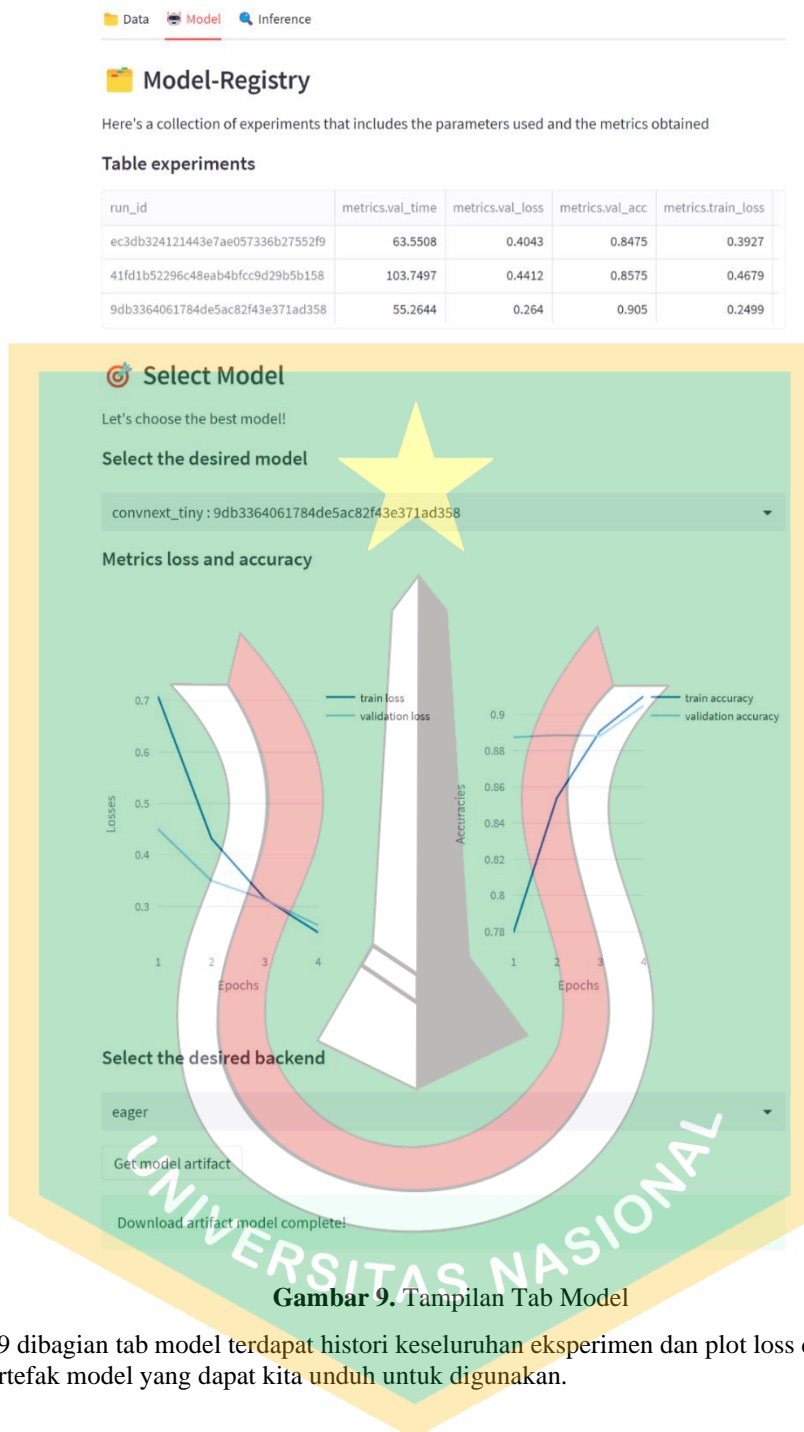
Gambar 7. Tampilan Tab Data pada Halaman Recognition

Pada Gambar 7 terdapat halaman recognition yang di dalamnya terdapat tab data, tempat dimana kita bisa memilih kelas yang diinginkan atau menambahkan kelas baru dan mengunggah gambar yang ingin kita klasifikasikan. Pada halaman labeling juga memiliki tampilan yang sama, hanya saja inputan gambar query dapat mengunggah lebih dari satu gambar.



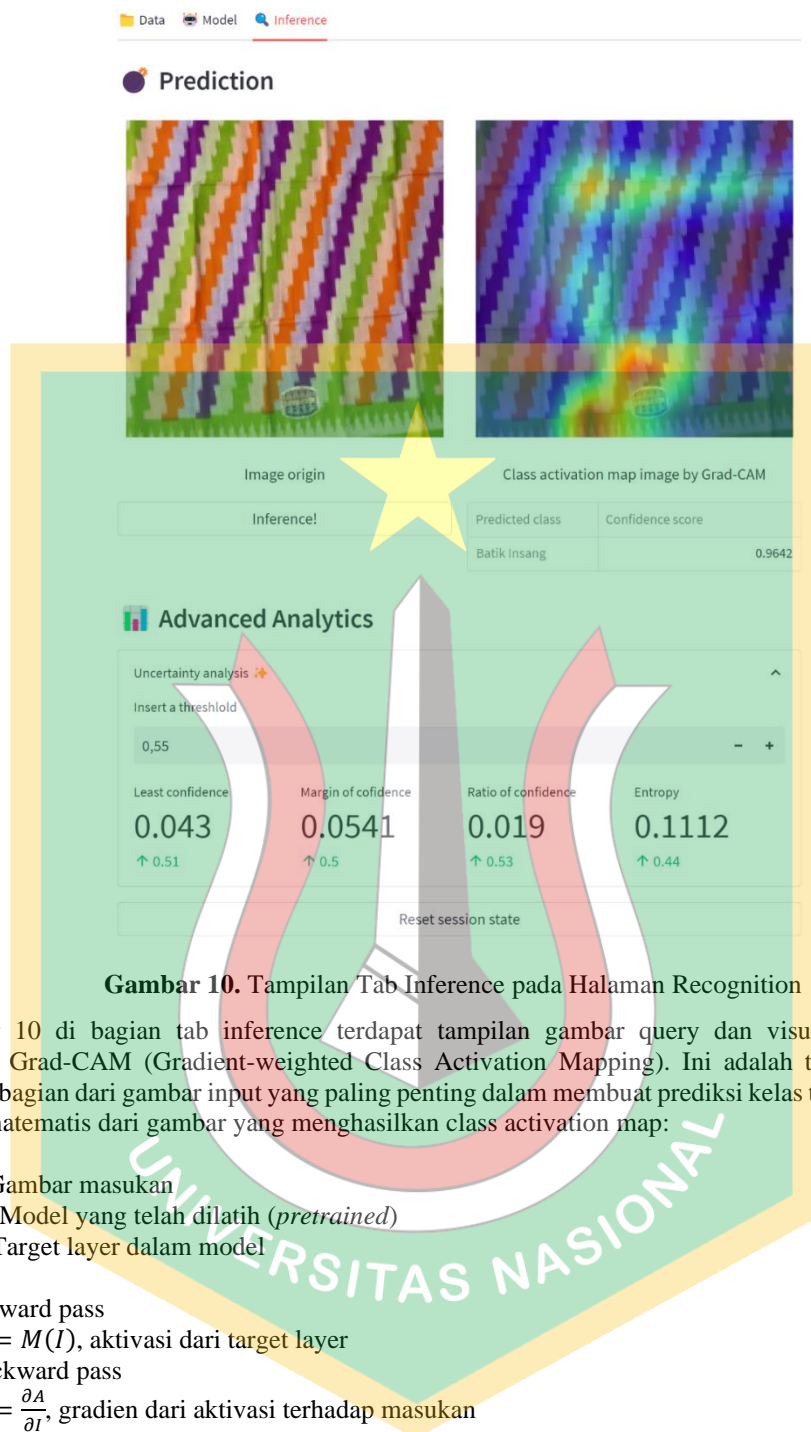
Gambar 8. Tampilan Tab Data dengan Delas Baru yang Ditambahkan

Pada Gambar 8 terdapat bagian support set, pengguna juga dapat menambahkan kelas baru dan mengunggah beberapa sampel gambar. Sebagai contoh, pada gambar diatas ditambahkan kelas baru yakni, “Batik Madagaskar”. Pengguna perlu mengunggah 10 sampel gambar-gambar Batik Madagaskar.



Gambar 9. Tampilan Tab Model

Pada Gambar 9 dibagian tab model terdapat histori keseluruhan eksperimen dan plot loss dan akurasi model yang dipilih, serta artefak model yang dapat kita unduh untuk digunakan.



Gambar 10. Tampilan Tab Inference pada Halaman Recognition

Pada Gambar 10 di bagian tab inference terdapat tampilan gambar query dan visualisasi activation map menggunakan Grad-CAM (Gradient-weighted Class Activation Mapping). Ini adalah teknik visualisasi yang menunjukkan bagian dari gambar input yang paling penting dalam membuat prediksi kelas tertentu. Berikut adalah representasi matematis dari gambar yang menghasilkan class activation map:

a. Input

- 1) I : Gambar masukan
- 2) M : Model yang telah dilatih (*pretrained*)
- 3) L : Target layer dalam model

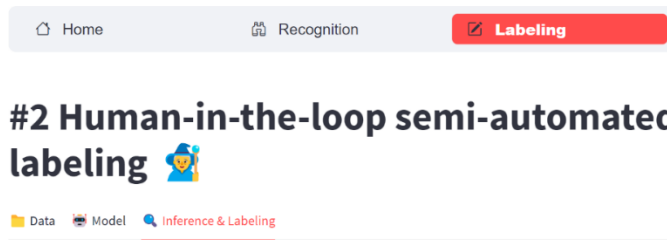
b. Proses

- 1) Forward pass
 $A = M(I)$, aktivasi dari target layer
- 2) Backward pass
 $G = \frac{\partial A}{\partial I}$, gradien dari aktivasi terhadap masukan
- 3) Perhitungan bobot
 $\alpha_k = \text{mean}(G, \text{dim} = (2, 3), \text{keepdim} = \text{True})$, bobot untuk kombinasi layer
- 4) Kombinasi linear
 $CAM = \sum \alpha_k \cdot A$, kombinasi linear dari aktivasi dengan bobot
- 5) Normalisasi dan ReLU
 $CAM = \text{ReLU}(CAM - \min(CAM)) / \max(CAM)$, normalisasi cam ke rentang [0,1]
- 6) Resize
 $CAM = \text{resize}(CAM, \text{size} = (\text{width}, \text{height}))$, ubah ukuran CAM ke ukuran gambar masukan

c. Output

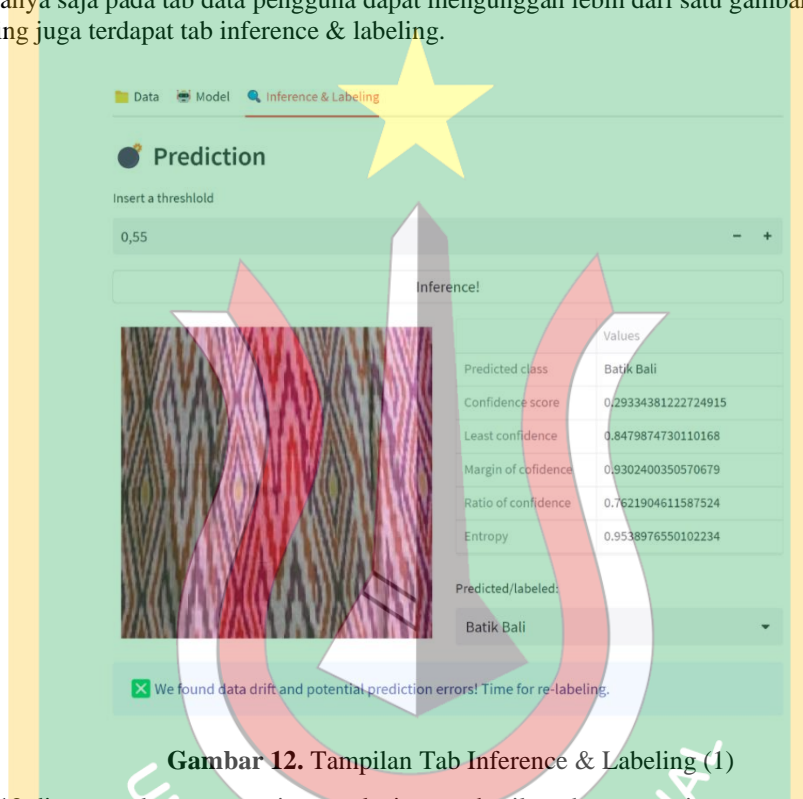
- 1) CAM: class activation map

Dengan menggunakan Grad-CAM, pengguna dapat melihat area spesifik dalam gambar yang menjadi fokus utama model saat membuat prediksi. Visualisasi ini dapat memberikan wawasan tambahan tentang bagaimana model menginterpretasikan motif batik dan membantu dalam analisis lebih lanjut tentang performa model [20].



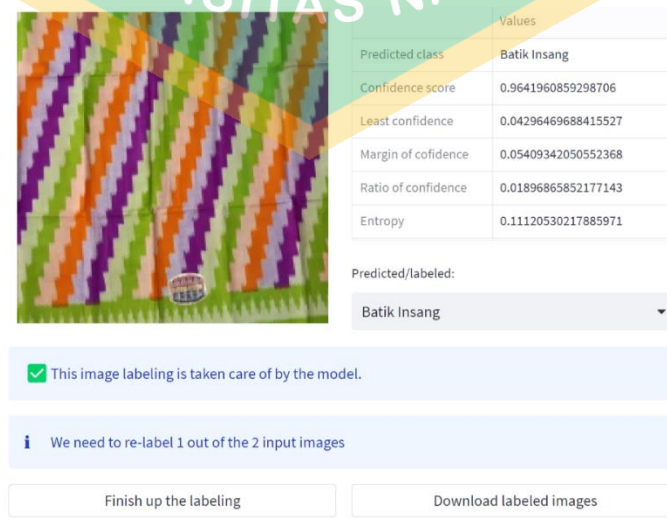
Gambar 11. Tampilan Gambar Header Halaman Labeling

Pada Gambar 10, halaman labeling juga terdapat tab data dan tab model. Kedua tab sama seperti pada halaman recognition. Hanya saja pada tab data pengguna dapat mengunggah lebih dari satu gambar. Selain itu pada halaman labeling juga terdapat tab inference & labeling.



Gambar 12. Tampilan Tab Inference & Labeling (1)

Pada Gambar 12 diatas, terdapat uncertainty analysis yang hasilnya berupa peringatan yang menunjukkan adanya data drift adanya potensi prediksi error. Dimana seharusnya itu adalah Batik Insang namun di prediksi Batik Bali. Pengguna perlu melakukan pelabelan ulang dengan memilih label yang benar pada select box.



Gambar 13. Tampilan Tab Inference & Labeling (2)



Gambar 13 adalah contoh pelabelan yang tidak perlu dilabeli ulang oleh manusia. Hasil uncertainty analysis menunjukkan pelabelan ini cukup ditangani oleh model saja tanpa interaksi manusia. Setelah itu terdapat informasi yang memberitahu pengguna bahwa dari dua gambar query terdapat satu gambar yang perlu interaksi manusia untuk pelabelan ulang. Pengguna juga dapat menyelesaikan tugas pelabelan dan mengunduh gambar yang telah dilabeli yang berupa file berformat .zip berisi folder yang sesuai dengan label gambar query.

4. KESIMPULAN

Berdasarkan penelitian ini, metode optimasi pengenalan dan pelabelan motif batik dengan Prototypical Networks dan Uncertainty Analysis telah membuka jalan baru dalam fleksibilitas pengenalan motif. Berbeda dengan metode klasifikasi konvensional yang memiliki keterbatasan dalam penambahan kelas baru, pendekatan ini memungkinkan penambahan kelas motif batik apapun tanpa perlu melatih ulang model, hanya dengan sedikit contoh. Hal ini terbukti dengan performa yang impresif saat diuji terhadap 6 novel kelas, dengan akurasi mencapai 92,63% dalam skenario 10-shot 3-way. Teknik pelabelan semi otomatis juga berhasil mengurangi biaya pelabelan sebanyak 76% dari total 120 data. Implementasi dalam web app melalui Streamlit dan Huggingface tidak hanya memungkinkan penggunaan yang lebih luas, tetapi juga menyediakan analitik lanjutan melalui uncertainty analysis. Dengan ini, pengguna dapat memahami bagaimana model membuat keputusan, mendapatkan wawasan berharga, dan bahkan melihat visualisasi Class Activation Map dengan Grad-CAM yang membantu dalam interpretasi model. Platform ini juga mendukung pelabelan kolaboratif antara model dan manusia, memungkinkan pelabelan langsung saat data drift terdeteksi atau potensi kesalahan prediksi. Keseluruhan, solusi ini tidak hanya menawarkan pendekatan yang kuat dan fleksibel untuk tantangan pengenalan dan pelabelan motif batik, tetapi juga berpotensi mendukung pelestarian dan promosi warisan budaya Indonesia dalam skala yang lebih besar.

REFERENCES

- [1] H. Ibda, "STRATEGI MEMUTUS MATA RANTAI PEMBAJAKAN HAK CIPTA PADA SENI BATIK NUSANTARA," 2021. [Online]. Available: <https://kumparan.com/haikalkurniawan/pentingnya->
- [2] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning", doi: 10.1007/s12525-021-00475-2/Published.
- [3] I. Martinez, E. Viles, and I. G. Olaizola, "A survey study of success factors in data science projects," Jan. 2022, doi: 10.1109/BigData52589.2021.9671588.
- [4] M. A. Rasyidi, R. Handayani, and F. Aziz, "Identification of batik making method from images using convolutional neural network with limited amount of data," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 3, pp. 1300–1307, Jun. 2021, doi: 10.11591/eei.v10i3.3035.
- [5] Universitas Bina Nusantara. School of Information Systems, Institute of Electrical and Electronics Engineers. Indonesia Section, and Institute of Electrical and Electronics Engineers, *Proceedings of 2019 International Conference on Information Management and Technology (ICIMTech) : date and venue: 19-20 August 2019, Hilton Hotel, Bali, & Bina Nusantara University, Jakarta, Indonesia*.
- [6] J. He, Z. Zhu, X. Fan, Y. Chen, S. Liu, and D. Chen, "Few-Shot Learning for Fault Diagnosis: Semi-Supervised Prototypical Network with Pseudo-Labels," *Symmetry (Basel)*, vol. 14, no. 7, Jul. 2022, doi: 10.3390/sym14071489.
- [7] S. Ackerman, E. Farchi, O. Raz, M. Zalmanovici, and P. Dube, "Detection of data drift and outliers affecting machine learning model performance over time," Dec. 2020, [Online]. Available: <http://arxiv.org/abs/2012.09258>
- [8] F. Shi, Z. Wang, M. Hu, and G. Zhai, "Active learning plus deep learning can establish cost-effective and robust model for multichannel image: A case on hyperspectral image classification," *Sensors (Switzerland)*, vol. 20, no. 17, MDPI AG, pp. 1–14, Sep. 01, 2020. doi: 10.3390/s20174975.
- [9] M. Hou and I. Sato, "A Closer Look at Prototype Classifier for Few-shot Image Classification," Oct. 2021, [Online]. Available: <http://arxiv.org/abs/2110.05076>
- [10] H. Prasetyo and B. A. Putra Akardihas, "Batik image retrieval using convolutional neural network," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 17, no. 6, pp. 3010–3018, Dec. 2019, doi: 10.12928/TELKOMNIKA.v17i6.12701.
- [11] J. C. Gomes and D. L. Borges, "Insect Pest Image Recognition: A Few-Shot Machine Learning Approach including Maturity Stages Classification," *Agronomy*, vol. 12, no. 8, Aug. 2022, doi: 10.3390/agronomy12081733.
- [12] G. Song, Z. Tao, X. Huang, G. Cao, W. Liu, and L. Yang, "Hybrid Attention-Based Prototypical Network for Unfamiliar Restaurant Food Image Few-Shot Recognition," *IEEE Access*, vol. 8, pp. 14893–14900, 2020, doi: 10.1109/ACCESS.2020.2964836.
- [13] H. Hino, "Active Learning: Problem Settings and Recent Developments," Dec. 2020, [Online]. Available: <http://arxiv.org/abs/2012.04225>
- [14] S. L. Hor *et al.*, "Deep Active Learning for Pornography Recognition Using ResNet," *International Journal of Technology*, vol. 13, no. 6, pp. 1261–1270, 2022, doi: 10.14716/ijtech.v13i6.5842.
- [15] F. Shi, Z. Wang, M. Hu, and G. Zhai, "Active learning plus deep learning can establish cost-effective and robust model for multichannel image: A case on hyperspectral image classification," *Sensors (Switzerland)*, vol. 20, no. 17, MDPI AG, pp. 1–14, Sep. 01, 2020. doi: 10.3390/s20174975.



- [16] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing Network Design Spaces," Mar. 2020, [Online]. Available: <http://arxiv.org/abs/2003.13678>
- [17] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," Apr. 2021, [Online]. Available: <http://arxiv.org/abs/2104.00298>
- [18] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," Jan. 2022, [Online]. Available: <http://arxiv.org/abs/2201.03545>
- [19] A. Parnami and M. Lee, "Learning from Few Examples: A Summary of Approaches to Few-Shot Learning," Mar. 2022, [Online]. Available: <http://arxiv.org/abs/2203.04291>
- [20] B. Zhang, B. Gao, S. Liang, X. Li, and H. Wang, "A classification algorithm based on improved meta learning and transfer learning for few-shot medical images," *IET Image Process*, 2023, doi: 10.1049/ipr2.12889.

