

BINUS
UNIVERSITY

BATIK PATTERN CLASSIFICATION

Kelompok:

Evan Fanuel - 2702327116

Vincent - 2702279325

Vincent Antonio Ekachandra - 2702240891

LATAR BELAKANG

Batik merupakan warisan budaya Indonesia yang memiliki nilai seni dan filosofi tinggi dengan beragam motif bermakna simbolik. Identifikasi batik secara manual memerlukan keahlian khusus dan berpotensi menimbulkan subjektivitas. Oleh karena itu, dikembangkan sistem klasifikasi otomatis menggunakan Convolutional Neural Network (CNN) dengan arsitektur ResNet50 melalui teknik transfer learning. Aplikasi berbasis Streamlit digunakan untuk memudahkan pengguna mengunggah citra batik dan memperoleh hasil klasifikasi secara cepat, interaktif, serta mendukung pelestarian dan edukasi budaya batik.

RUMUSAN MASALAH

Batik merupakan warisan budaya Indonesia yang memiliki nilai seni dan filosofi tinggi dengan beragam motif bermakna simbolik. Identifikasi batik secara manual memerlukan keahlian khusus dan berpotensi menimbulkan subjektivitas. Oleh karena itu, dikembangkan sistem klasifikasi otomatis menggunakan Convolutional Neural Network (CNN) dengan arsitektur ResNet50 melalui teknik transfer learning. Aplikasi berbasis Streamlit digunakan untuk memudahkan pengguna mengunggah citra batik dan memperoleh hasil klasifikasi secara cepat, interaktif, serta mendukung pelestarian dan edukasi budaya batik.

Dataset Link : <https://www.kaggle.com/datasets/hendryhb/batik-nusantara-batik-indonesia-dataset>

DATASET

PREVIEW IMAGE



DATASET HANDLING

```
IMG_SIZE = (224, 224)
BATCH_SIZE = 16
EPOCHS = 50
NUM_CLASSES = 20

TRAIN_DIR = "./batik_dataset/train/"
TEST_DIR = "./batik_dataset/test/"
```

```
train_ds = image_dataset_from_directory(
    TRAIN_DIR,
    image_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    label_mode="categorical",
    shuffle=True
)

test_ds = image_dataset_from_directory(
    TEST_DIR,
    image_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    label_mode="categorical",
    shuffle=False
)

class_names = train_ds.class_names
print("Classes:", class_names)
```

```
AUTOTUNE = tf.data.AUTOTUNE
train_ds = train_ds.prefetch(buffer_size=AUTOTUNE)
test_ds = test_ds.prefetch(buffer_size=AUTOTUNE)
```

DATA PREPROCESS

```
data_augmentation = tf.keras.Sequential([
    layers.RandomFlip("horizontal"),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
    layers.RandomContrast(0.1),
])
```

MODEL (RESNET50)

Model yang digunakan dalam proyek ini adalah ResNet50 dengan pendekatan transfer learning dan fine-tuning, yang disimpan dalam file batik_classification_model.keras. ResNet50 dipilih karena kemampuannya dalam mengekstraksi fitur visual yang kompleks melalui arsitektur deep residual network. Dengan memanfaatkan bobot awal dari dataset ImageNet, model dapat mengenali pola dasar seperti tekstur, bentuk, dan warna batik secara lebih efektif meskipun jumlah dataset terbatas. Proses fine-tuning dilakukan dengan membuka beberapa layer terakhir untuk dilatih ulang agar model lebih adaptif terhadap karakteristik khusus motif batik. Informasi kelas disimpan dalam file class_names.npy, yang berfungsi sebagai pemetaan antara output model dan label kelas.

MODEL (RESNET50)

```
base_model = ResNet50(  
    weights="imagenet",  
    include_top=False,  
    input_shape=(224, 224, 3)  
)  
  
base_model.trainable = False # Freeze backbone  
  
model.compile(  
    optimizer=Adam(learning_rate=1e-4),  
    loss="categorical_crossentropy",  
    metrics=["accuracy"]  
)  
  
model.summary()
```

```
inputs = layers.Input(shape=(224, 224, 3))  
x = data_augmentation(inputs)  
x = tf.keras.applications.resnet50.preprocess_input(x)  
  
x = base_model(x, training=False)  
x = layers.GlobalAveragePooling2D()(x)  
x = layers.BatchNormalization()(x)  
x = layers.Dense(256, activation="relu")(x)  
x = layers.Dropout(0.5)(x)  
  
outputs = layers.Dense(NUM_CLASSES, activation="softmax")(x)  
  
model = models.Model(inputs, outputs)
```

MODEL ARCHITECTURE

Layer (type)	Output Shape	Param #	Connected to
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0	-
sequential (Sequential)	(None, 224, 224, 3)	0	input_layer_1[0]...
get_item (GetItem)	(None, 224, 224)	0	sequential[0][0]
get_item_1 (GetItem)	(None, 224, 224)	0	sequential[0][0]
get_item_2 (GetItem)	(None, 224, 224)	0	sequential[0][0]
stack (Stack)	(None, 224, 224, 3)	0	get_item[0][0], get_item_1[0][0], get_item_2[0][0]
add (Add)	(None, 224, 224, 3)	0	stack[0][0]
resnet50 (Functional)	(None, 7, 7, 2048)	23,587,712	add[0][0]
global_average_poo... (GlobalAveragePool...)	(None, 2048)	0	resnet50[0][0]
batch_normalization (BatchNormalizatio...)	(None, 2048)	8,192	global_average_p...
dense (Dense)	(None, 256)	524,544	batch_normalizat...
dropout (Dropout)	(None, 256)	0	dense[0][0]
dense_1 (Dense)	(None, 20)	5,140	dropout[0][0]

MODEL TRAINING

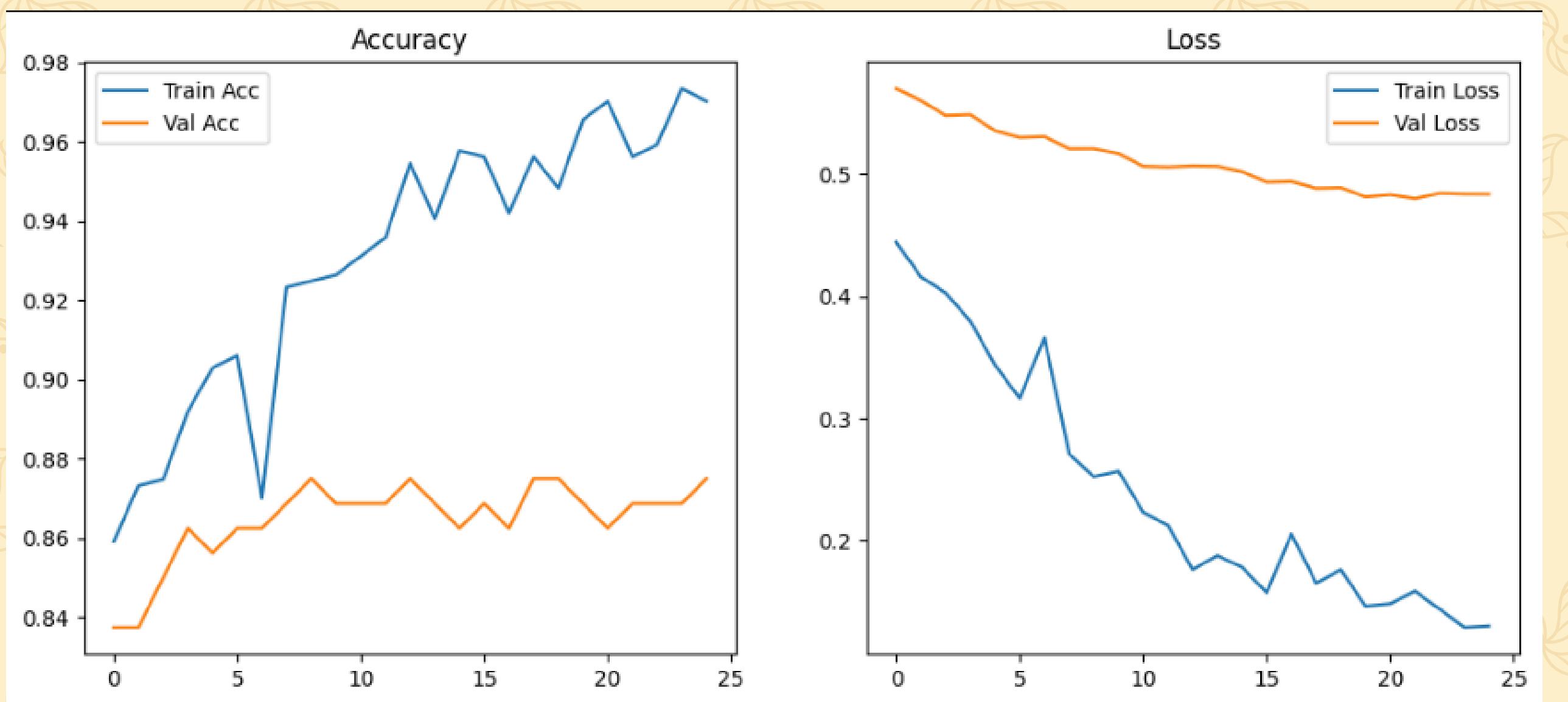
Training

```
early_stopping = tf.keras.callbacks.EarlyStopping(  
    monitor="val_loss",  
    patience=5,  
    restore_best_weights=True  
)  
  
history = model.fit(  
    train_ds,  
    validation_data=test_ds,  
    epochs=EPOCHS,  
    callbacks=[early_stopping]  
)
```

Fine Tuned

```
# Unfreeze model  
base_model.trainable = True  
  
# Freeze sebagian layer (agar tidak overfitting)  
# unfreeze 30 layer terakhir  
for layer in base_model.layers[:-30]:  
    layer.trainable = False  
  
model.compile(  
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-5),  
    loss="categorical_crossentropy",  
    metrics=["accuracy"]  
)  
  
fine_tune_history = model.fit(  
    train_ds,  
    validation_data=test_ds,  
    epochs=25,  
    callbacks=[early_stopping])
```

GRAPHS



EVALUATION

Evaluasi model dilakukan secara ketat menggunakan data uji (test) yang terpisah dari data latih, memastikan kemampuan generalisasi model dan menghindari overfitting. Detail pelatihan, validasi, dan pengujian performa, termasuk analisis akurasi dan kesalahan klasifikasi, didokumentasikan sepenuhnya dalam file ProjectDL.ipynb

MODEL EVALUATION

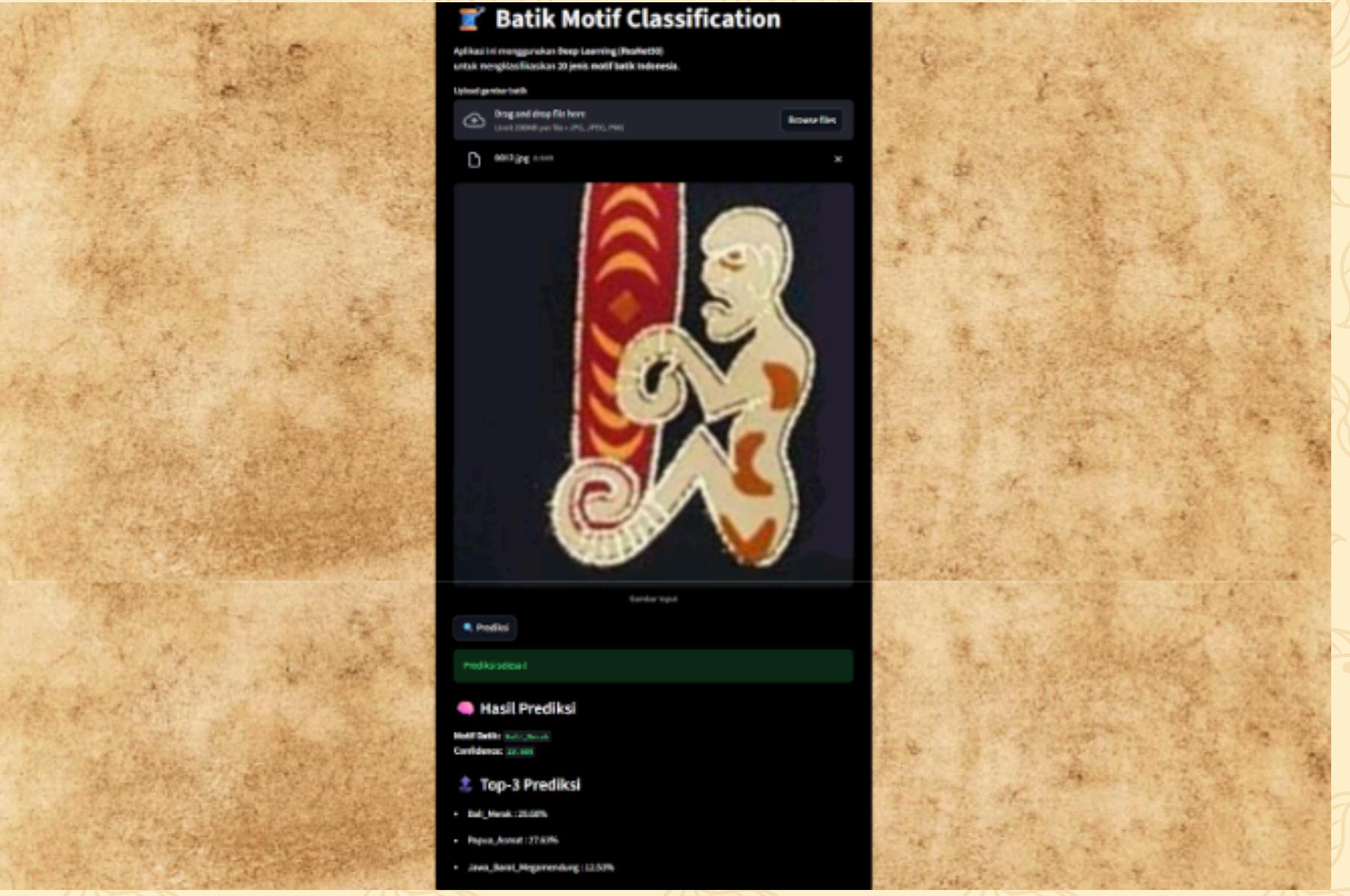
```
final_test_loss, final_test_acc = model.evaluate(test_ds)
print(f"Final Test Accuracy : {final_test_acc:.4f}")

10/10 ━━━━━━━━━━ 15s 1s/step - accuracy: 0.8687 - loss: 0.4800
Final Test Accuracy : 0.8687
```

CLASSIFICATION REPORT

	precision	recall	f1-score	support
Aceh_Pintu_Aceh	0.60	0.38	0.46	8
Bali_Barong	0.89	1.00	0.94	8
Bali_Merak	0.89	1.00	0.94	8
DKI_Ondel_Ondel	1.00	0.75	0.86	8
Jawa_Barat_Megamendung	1.00	0.88	0.93	8
Jawa_Timur_Pring	0.78	0.88	0.82	8
Kalimantan_Dayak	1.00	1.00	1.00	8
Lampung_Gajah	0.79	0.88	0.78	8
Madura_Mataketeran	0.86	0.75	0.80	8
Maluku_Pala	0.80	1.00	0.89	8
NTB_Lumbung	0.86	0.75	0.80	8
Papua_Asmat	0.80	1.00	0.89	8
Papua_Cendrawasih	1.00	0.75	0.86	8
Papua_Tifa	0.88	0.88	0.88	8
Solo_Parang	0.78	0.88	0.82	8
Sulawesi_Selatan_Lontara	1.00	0.88	0.93	8
Sumatera_Barat_Rumah_Minang	1.00	0.88	0.93	8
Sumatera_Utara_Boraspati	0.80	1.00	0.89	8
Yogyakarta_Kawung	1.00	1.00	1.00	8
Yogyakarta_Parang	0.88	0.88	0.88	8
accuracy			0.87	160
macro avg	0.87	0.87	0.86	160
weighted avg	0.87	0.87	0.86	160

DEPLOYMENT

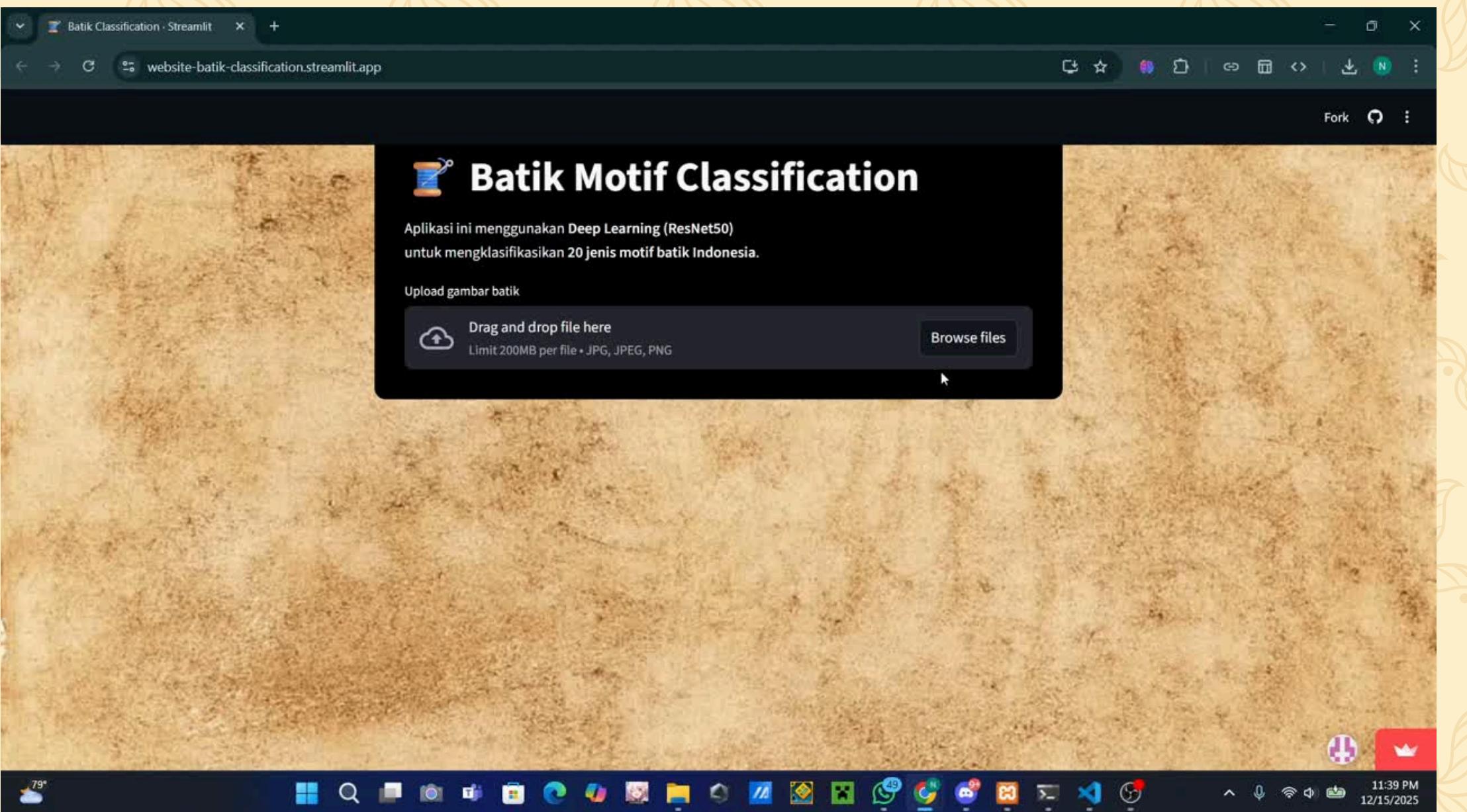


Link Deployment : <https://website-batik-classification.streamlit.app/>

Link Github:

<https://github.com/aidennz/Batik-Classification.git>

VIDEO DEMO



Link Video :

[https://drive.google.com/drive/folders/1B0Zi77TYbyy
PVbsOkzWgWQq3c240m_co?usp=sharing](https://drive.google.com/drive/folders/1B0Zi77TYbyyPVbsOkzWgWQq3c240m_co?usp=sharing)

KESIMPULAN

Proyek ini berhasil mengintegrasikan penuh Deep Learning Pipeline untuk klasifikasi citra batik. Kami menggunakan ResNet50 dengan teknik Fine-tuning yang efektif. Sistem ini telah di-deploy menggunakan Streamlit, menghasilkan solusi yang aplikatif dan mudah diakses.

Pengembangan selanjutnya akan berfokus pada peningkatan variasi data dan penggunaan metrik evaluasi yang lebih detail.

PENUTUP



**TERIMAKASIH TELAH
MENGINFOKAN**