

MULTICLASS IMAGE CLASSIFICATION OF NATURAL DISASTERS USING CNN

Crista Livia Budiman / 2702271133

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi AI memberikan berbagai dampak besar bagi kehidupan manusia, salah satunya adalah penggunaan teknologi Deep Learning, khususnya Convolutional Neural Network (CNN) yang mengolah gambar untuk diidentifikasi dan melakukan pengklasifikasian dengan tingkat akurasi yang tinggi. Dalam project ini, CNN bermanfaat untuk mengklasifikasikan bencana berdasarkan gambar-gambar. Akan tetapi, terdapat beberapa tantangan dalam melakukan klasifikasi gambar bencana, seperti ketidakseimbangan data dan kompleksitas visual pada gambar.

1.2 Penjelasan Mengenai Problem

Masalah utama dalam project ini adalah kebutuhan agar model dapat secara otomatis mengklasifikasikan gambar-gambar bencana secara efektif dan akurat ke dalam kategori tertentu. Tantangan utama yang dihadapi meliputi ketidakseimbangan jumlah data antar kelas dan kompleksitas visual gambar bencana yang dapat mempersulit proses klasifikasi.

1.3 Task yang Dilakukan

Project ini melakukan *image classification*, yaitu mengidentifikasi dan mengelompokkan gambar-gambar yang tersedia ke dalam kategori tertentu. Metode utama yang digunakan adalah CNN, yang dioptimalkan melalui tahap preprocessing seperti augmentasi data dan normalisasi. Proses ini dilanjutkan dengan training model, validasi, dan evaluasi performa menggunakan metrik seperti accuracy dan *confusion matrix* untuk mengukur keefisienan model dan keberhasilan klasifikasi.

1.4 Hasil Analisis Sederhana Terhadap Data

Dataset yang digunakan terdiri atas 6 kategori utama, yaitu:

- a. Damaged Infrastructure (Earthquake, Infrastructure): gambar kerusakan infrastruktur atau dampak gempa.

- b. Fire Disaster (Urban Fire, Wild Fire): gambar kebakaran di area urban dan hutan.
- c. Human Damage: gambar dampak bencana terhadap manusia.
- d. Land Disaster (Drought, Land Slide): gambar bencana tanah seperti kekeringan dan longsor.
- e. Non Damage (Non Damage Buildings Street, Non Damage Wildfire Forest, human, sea): gambar tanpa kerusakan, seperti bangunan dan hutan utuh, manusia, dan perairan.
- f. Water Disaster: gambar bencana yang disebabkan oleh air.

Untuk project multiclass image classification ini, terdapat 4 kelas yang akan digunakan, yaitu Earthquake (36 gambar), Urban Fire (419 gambar), Land Slide (456 gambar), dan Water Disaster (1035 gambar). Berdasarkan analisis pada tahap data preparation, ditemukan ketidakseimbangan jumlah data antar kelas, dengan jumlah gambar terbanyak pada kelas Water Disaster sebanyak 1035 gambar dan paling sedikit pada kelas Earthquake sebanyak 36 gambar. Ketidakseimbangan ini dapat berdampak negatif terhadap kinerja model klasifikasi yang akan dikembangkan. Oleh karena itu, diperlukan pendekatan khusus seperti augmentasi data dan penyesuaian bobot kelas (class weight) untuk mengatasi ketidakseimbangan tersebut.

II. METODOLOGI

2.1 Pendekatan Penyelesaian Masalah

Penelitian ini mengadopsi pendekatan Deep Learning dengan memanfaatkan arsitektur Convolutional Neural Network (CNN) untuk menyelesaikan permasalahan klasifikasi citra bencana alam yang bersifat multiclass. Proses pengembangan model dilakukan secara bertahap, dimulai dari eksplorasi dan persiapan data, pelatihan model, hingga evaluasi akhir performa. Mengingat adanya ketimpangan jumlah data antar kelas yang cukup signifikan, pendekatan ini juga difokuskan pada strategi penanganan data tidak seimbang agar hasil klasifikasi menjadi lebih akurat dan adil terhadap semua kategori.

2.2 Tahapan Penelitian

a. Dataset dan Karakteristik Gambar

Dataset yang digunakan dalam penelitian ini memuat gambar-gambar dari empat kategori utama bencana alam, dengan total keseluruhan sebanyak 1.946 gambar, yang terbagi sebagai berikut:

- Earthquake: 36 gambar ($\pm 1,8\%$)
- Urban Fire: 419 gambar ($\pm 21,4\%$)
- Land Slide: 456 gambar ($\pm 23,3\%$)
- Water Disaster: 1035 gambar ($\pm 53,5\%$)

Distribusi ini menunjukkan adanya ketidakseimbangan yang cukup ekstrem, terutama antara kelas Earthquake dan Water Disaster, dengan rasio hingga 1:28. Ketimpangan seperti ini berpotensi menyebabkan model terlalu fokus pada kelas mayoritas, sehingga perlu ditangani dengan strategi khusus.

b. Persiapan dan Pra-pemrosesan Data

Sebelum melatih model, dilakukan tahapan *data preparation* untuk mengenali pola distribusi dan mengantisipasi ketidakseimbangan antar kelas. Dua strategi utama yang diterapkan dalam tahap ini antara lain:

- Analisis Distribusi Data: Visualisasi dan perhitungan statistik digunakan untuk melihat distribusi jumlah gambar di tiap kelas secara menyeluruh.
- Penanganan Ketimpangan Data:
 - *Data Augmentation*: Teknik seperti rotasi, pemotongan (*cropping*), pencahayaan ulang, pembesaran (*zooming*), serta pembalikan gambar (*flipping*) diterapkan khusus pada kelas minoritas guna memperkaya variasi data tanpa menambah data baru secara manual.
 - *Class Weight Adjustment*: Menyesuaikan bobot masing-masing kelas saat proses pelatihan agar model tidak bias terhadap kelas mayoritas.

c. Rancangan Arsitektur Model

Model yang digunakan mengacu pada arsitektur CNN dengan struktur bertingkat untuk mengekstrak fitur visual dari gambar secara bertahap. Komponen-komponen utama model mencakup:

- Layer Konvolusi: Menangkap pola dan fitur penting dari gambar input.
- Pooling Layer: Mengurangi dimensi data untuk efisiensi komputasi.
- Dropout Layer: Digunakan untuk mencegah overfitting dengan mengabaikan beberapa neuron saat pelatihan.

- Fully Connected (Dense) Layer: Menyusun hasil akhir klasifikasi ke dalam 4 output kelas sesuai dengan kategori bencana yang ditetapkan.

d. Proses Pelatihan Model

Dataset dibagi ke dalam tiga bagian dengan proporsi berikut:

- Data latih (Training Set): 70%
- Data validasi (Validation Set): 15%
- Data uji (Test Set): 15%

Selama proses pelatihan, dilakukan *tuning* pada beberapa parameter seperti *learning rate*, *batch size*, jumlah *epoch*, serta pemilihan *optimizer* terbaik untuk mencapai performa model yang optimal.

e. Evaluasi dan Pengukuran Kinerja Model

Setelah pelatihan, performa model dievaluasi menggunakan berbagai metrik yang mencerminkan keakuratan dan ketepatan klasifikasi:

- Accuracy: Mengukur seberapa sering prediksi model benar secara keseluruhan.
- Precision: Menggambarkan akurasi prediksi positif terhadap jumlah prediksi positif.
- Recall: Menilai kemampuan model dalam mengenali seluruh data positif dari masing-masing kelas.
- F1-Score: Kombinasi harmonis dari precision dan recall untuk mendapatkan keseimbangan kinerja.
- Confusion Matrix: Memberikan gambaran lebih mendetail mengenai prediksi benar dan salah di setiap kelas.

f. Perangkat dan Framework yang Digunakan

Dalam implementasinya, penelitian ini menggunakan beberapa alat bantu pemrograman dan framework berikut:

- Python: Bahasa pemrograman utama dalam keseluruhan eksperimen.
- TensorFlow/Keras: Framework inti untuk membangun dan melatih model CNN.
- OpenCV: Digunakan untuk pemrosesan dan manipulasi gambar.
- Scikit-learn: Mendukung proses evaluasi performa model.

- Matplotlib: Untuk visualisasi data dan hasil evaluasi model.

III. HASIL DAN ANALISIS

3.1 Desain Eksperimen dan Variasi Model

Penelitian ini melibatkan beberapa konfigurasi model CNN yang dibangun untuk menyelesaikan tugas klasifikasi gambar bencana alam secara *multiclass*. Setiap eksperimen bertujuan untuk mengamati pengaruh penerapan strategi penanganan data tidak seimbang terhadap kinerja klasifikasi. Terdapat dua versi model yang dikembangkan dan diuji:

- Model 1: CNN dasar tanpa augmentasi data maupun penyesuaian bobot kelas.
- Model 2 (Model Optimal): CNN yang dikombinasikan dengan teknik augmentasi pada kelas minoritas dan penerapan class weighting selama pelatihan.

Untuk mendukung pemahaman terhadap struktur model, berikut adalah ilustrasi potongan kode yang menunjukkan arsitektur CNN yang digunakan:

```
def design_model1(input_shape=(224, 224, 3), num_classes=4):
    model = Sequential([
        Input(shape=input_shape),
        Conv2D(32, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),
        BatchNormalization(),

        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),
        BatchNormalization(),

        Conv2D(128, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),
        BatchNormalization(),

        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.5),
        Dense(num_classes, activation='softmax')
    ])
    return model
```

```
def design_model2(input_shape=(224, 224, 3), num_classes=4):
    base_model = MobileNetV2(include_top=False, weights='imagenet', input_shape=input_shape)
    base_model.trainable = True

    for layer in base_model.layers[:-50]:
        layer.trainable = False

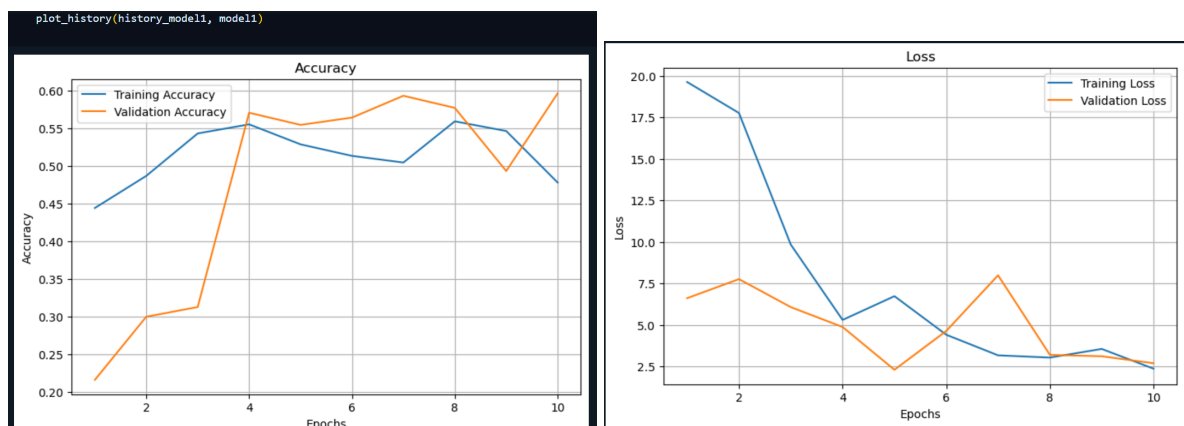
    x = base_model.output
    x = GlobalAveragePooling2D()(x)
    x = Dropout(0.3)(x)
    x = Dense(128, activation='relu')(x)
    x = Dropout(0.3)(x)
    predictions = Dense(num_classes, activation='softmax')(x)

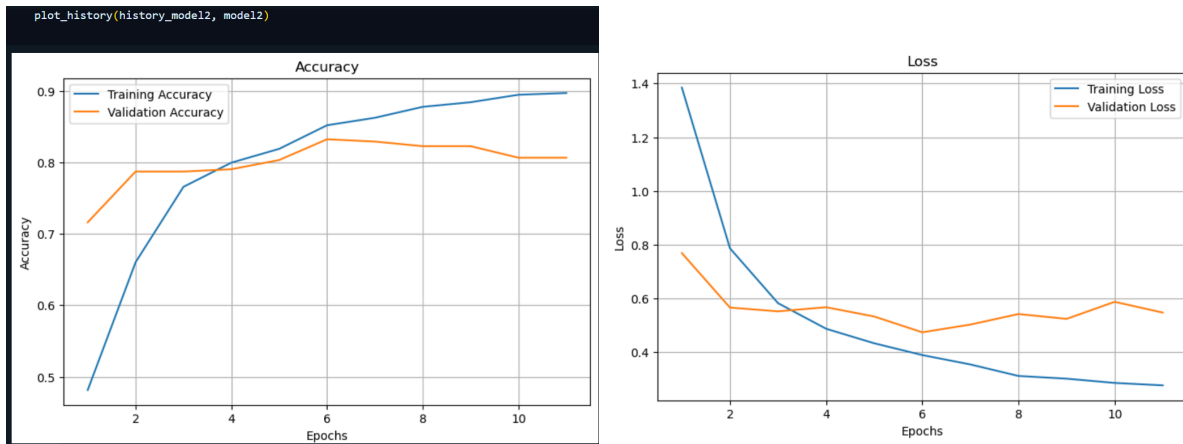
    model = Model(inputs=base_model.input, outputs=predictions)
    return model
```

Gambar 1. Cuplikan Kode Arsitektur Model CNN

3.2 Proses Pelatihan dan Validasi

Pada model pertama, pelatihan dilakukan tanpa penyesuaian terhadap distribusi data. Grafik akurasi dan loss menunjukkan bahwa model mengalami overfitting, di mana akurasi pelatihan meningkat, tetapi akurasi validasi stagnan dan fluktuatif. Sebaliknya, model kedua menunjukkan tren pelatihan yang lebih stabil, dengan konvergensi pada akurasi validasi yang lebih tinggi dan loss yang menurun secara konsisten. Hasil ini diperoleh berkat penerapan augmentasi citra dan class weight, yang membantu model mempelajari pola dari kelas minoritas secara lebih efektif. Berikut adalah visualisasi kurva akurasi dan loss dari proses pelatihan kedua model:





Gambar 2. Plot Akurasi dan Loss untuk Model 1 dan Model 2

3.3 Evaluasi Kinerja Model

Model terbaik, yaitu Model 2, menunjukkan performa yang cukup tinggi pada data uji, dengan hasil matrik sebagai berikut:

- Accuracy: 86,15%
- Precision (macro average): 84,53%
- Recall (macro average): 85,11%
- F1-Score (macro average): 84,81%

Nilai-nilai tersebut mencerminkan kemampuan model dalam melakukan prediksi secara seimbang antar kelas, sekalipun dengan distribusi data yang tidak merata.

```
def evaluate(model, test_ds, label_names):
    y_true = []
    y_pred = []
    for x, y in test_ds:
        pred = model.predict(x)
        y_pred.extend(np.argmax(pred, axis=1))
        y_true.extend(np.argmax(y.numpy(), axis=1))

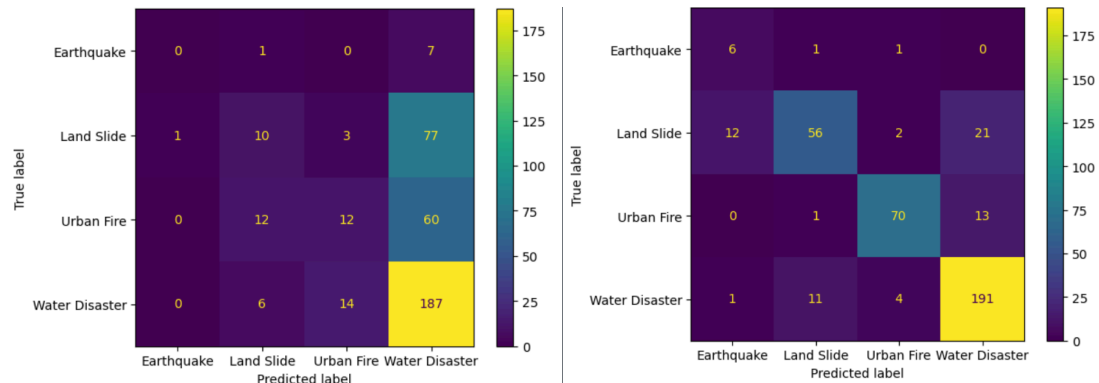
    print(classification_report(y_true, y_pred, target_names=label_names))
    cm = confusion_matrix(y_true, y_pred)
    ConfusionMatrixDisplay(cm, display_labels=label_names).plot()
```

Gambar 3. Cuplikan Kode Evaluasi Model dengan Metrik Sklearn

3.4 Analisis Confusion Matrix

Untuk memahami lebih jauh performa prediktif pada setiap kelas, digunakan confusion matrix. Hasil analisis menunjukkan bahwa model mampu mengklasifikasikan kelas Water

Disaster dengan sangat baik, namun masih terjadi sejumlah kesalahan klasifikasi pada kelas Earthquake yang cenderung dikira sebagai Urban Fire atau Land Slide.

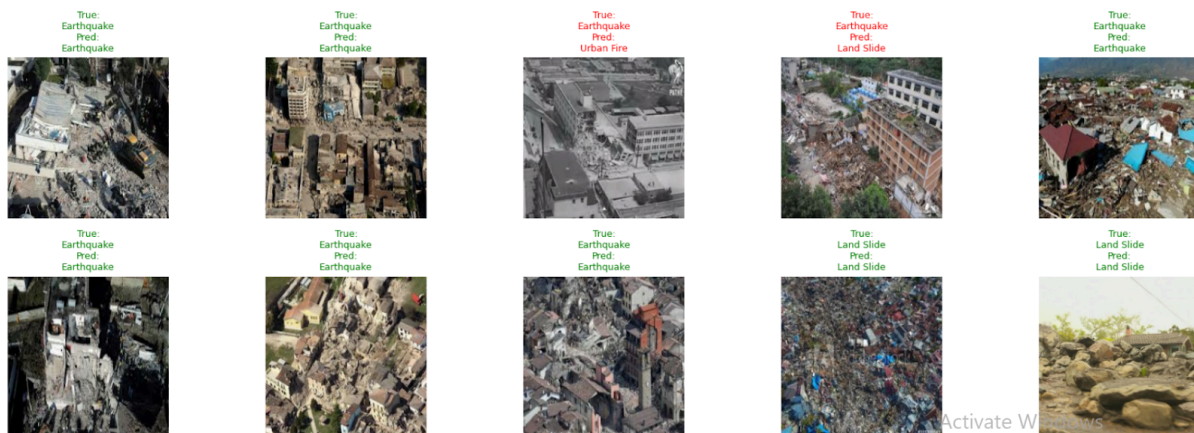


Gambar 4. Visualisasi Confusion Matrix Model Terbaik

Dari hasil tersebut dapat dilihat bahwa dominasi jumlah data pada kelas Water Disaster memberikan keuntungan dalam akurasi, sedangkan kelas dengan jumlah data sedikit seperti Earthquake masih menantang untuk diklasifikasi secara akurat.

3.5 Visualisasi Hasil Prediksi

Sebagai bagian dari evaluasi kualitatif, beberapa contoh hasil prediksi model divisualisasikan bersama dengan label sebenarnya. Hal ini dilakukan untuk melihat apakah model dapat mengenali fitur-fitur visual secara intuitif.



Gambar 5. Contoh Gambar Asli dan Prediksi Model

Dari hasil observasi visual, model cukup andal dalam memprediksi gambar Water Disaster dan Land Slide. Akan tetapi, untuk gambar-gambar Earthquake, model kerap mengalami misklasifikasi, yang mengindikasikan perlunya data tambahan atau strategi ekstraksi fitur yang lebih kompleks.

IV. KESIMPULAN

4.1 Rangkuman Hasil Penelitian

Penelitian ini menerapkan metode *Deep Learning* berbasis Convolutional Neural Network (CNN) untuk menyelesaikan persoalan klasifikasi gambar bencana alam yang melibatkan lebih dari satu kategori (*multiclass*). Melalui rangkaian eksperimen, ditemukan bahwa tantangan utama terletak pada distribusi data yang tidak seimbang antar kelas serta kerumitan visual dari citra bencana itu sendiri.

Dua model dikembangkan dan diuji untuk melihat dampak dari penerapan augmentasi data serta penyesuaian bobot kelas (*class weight*). Hasilnya menunjukkan bahwa model dengan pendekatan penyeimbangan data secara eksplisit mampu memberikan performa klasifikasi yang lebih unggul. Model terbaik mencapai tingkat akurasi sebesar 86,15% dan nilai F1-score sebesar 84,81%, yang mencerminkan keseimbangan yang baik antara presisi dan sensitivitas dalam pengenalan setiap kelas.

4.2 Interpretasi dan Evaluasi Kinerja

Temuan dari hasil penelitian ini menunjukkan beberapa poin penting:

- Penggunaan teknik augmentasi berhasil meningkatkan keberagaman data pada kelas yang memiliki jumlah terbatas, sehingga mendukung proses pembelajaran model.
- Penyesuaian bobot kelas saat pelatihan membuat model tidak terlalu bias terhadap kelas mayoritas, memungkinkan distribusi prediksi yang lebih adil.
- Hasil confusion matrix menunjukkan bahwa model masih mengalami kesulitan dalam membedakan beberapa kelas yang memiliki kemiripan visual, terutama antara Earthquake dan Urban Fire.

4.3 Keterbatasan Penelitian dan Rekomendasi

Meskipun secara keseluruhan model menunjukkan kinerja yang baik, terdapat beberapa batasan yang mempengaruhi hasil:

- Jumlah data pada kelas Earthquake sangat terbatas, yaitu hanya 36 gambar, yang menyebabkan akurasi prediksi terhadap kelas ini menjadi kurang optimal.
- Penelitian ini belum mengeksplorasi arsitektur CNN yang lebih kompleks seperti ResNet, VGG, atau EfficientNet, yang kemungkinan besar dapat meningkatkan hasil klasifikasi secara keseluruhan.

Untuk pengembangan lebih lanjut, beberapa saran yang dapat dipertimbangkan antara lain:

- Menambah jumlah data pada kelas minoritas agar distribusi dataset menjadi lebih proporsional.
- Melakukan eksplorasi terhadap arsitektur model yang lebih dalam dan kompleks untuk meningkatkan kapasitas generalisasi.
- Menerapkan teknik interpretabilitas model seperti Grad-CAM agar proses klasifikasi dapat lebih mudah dianalisis dan dipahami dari sisi visual.

REFERENSI

- Chollet, F. (2017). *Deep learning with Python*. Manning Publications.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4510–4520). <https://doi.org/10.1109/CVPR.2018.00474>
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1–48. <https://doi.org/10.1186/s40537-019-0197-0>
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/1409.1556>
- TensorFlow Developers. (2024). *Image classification*. TensorFlow. <https://www.tensorflow.org/tutorials/images/classification>