

Classification de Scènes Naturelles

Septembre 2025

“*Projet de Deep Learning*”



KOTO Patricia

Plan

1. Motivation et Objectifs

2. Jeu de données

3. Méthodologie

4. Modélisation

5. Résultats

6. Appllcation Streamlit

7. Conclusion et
perspectives

7 points



Motivation et objectifs

Pourquoi classifier automatiquement des scènes ?

- Organisation d'images
- Robotique,
- géographie,
- environnement

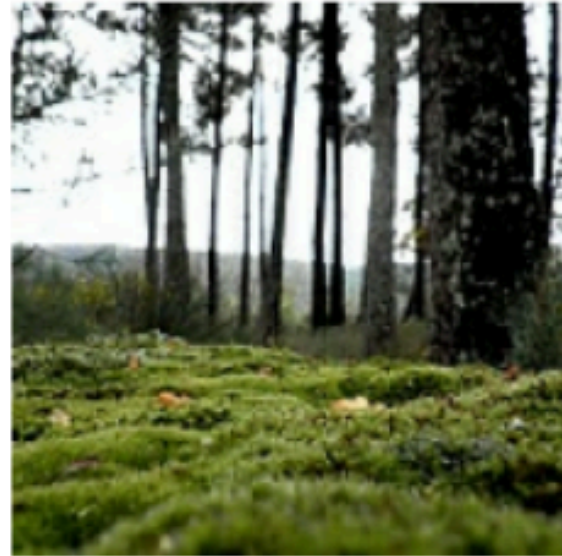
Objectif : prédire 6 classes (buildings, forest, glacier, mountain, sea, street)

Jeu de données

glacier



forest



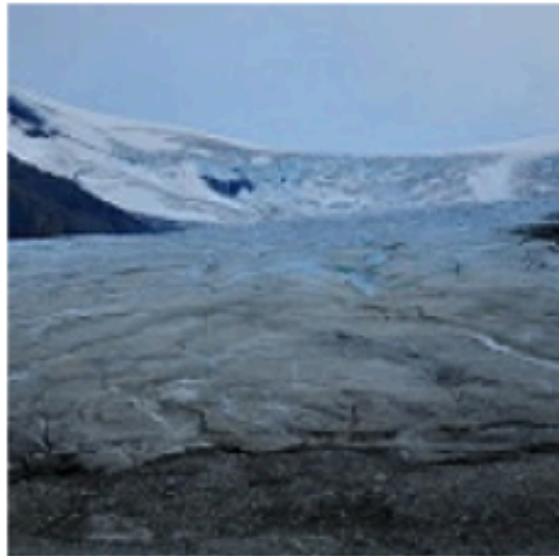
sea



street



glacier



mountain



glacier



buildings



Résultats du chargement :

- 14 034 images trouvées dans le dossier d'entraînement
- 3 000 images trouvées dans le dossier de validation
- 6 classes détectées : ['buildings', 'forest', 'glacier', 'mountain', 'sea', 'street'] Nous avons donc un problème de classification multi-classes avec 6 catégories.

**Proportions : {'buildings': '14.6%',
'forest': '15.8%',
'glacier': '18.4%',
'mountain': '17.5%',
'sea': '17.0%',
'street': '16.7%'}**

Pipeline CNN

1. **Chargement des données** → train, validation, test.
2. **Prétraitement & augmentation** → redimensionnement, normalisation, zoom, rotation, flip.
3. **Construction du modèle** → convolution → pooling → dense → softmax.
4. **Compilation** → fonction de perte (categorical crossentropy), optimiseur (Adam), métriques.
5. **Entraînement (fit)** → apprentissage sur train, suivi sur validation.
6. **Évaluation** → test, matrice de confusion, accuracy, précision, rappel, F1-score.
7. **Interprétation** → Grad-CAM pour visualiser les zones activées et analyser les erreurs.
8. **Améliorations** → tuning hyperparamètres, régularisation, transfert learning.

Méthodologie

- Préparation : normalisation, data augmentation
- Split : train / validation / test
- Modèles testés :
 - CNN baseline
 - Transfer Learning (MobileNetV2)

Modélisation

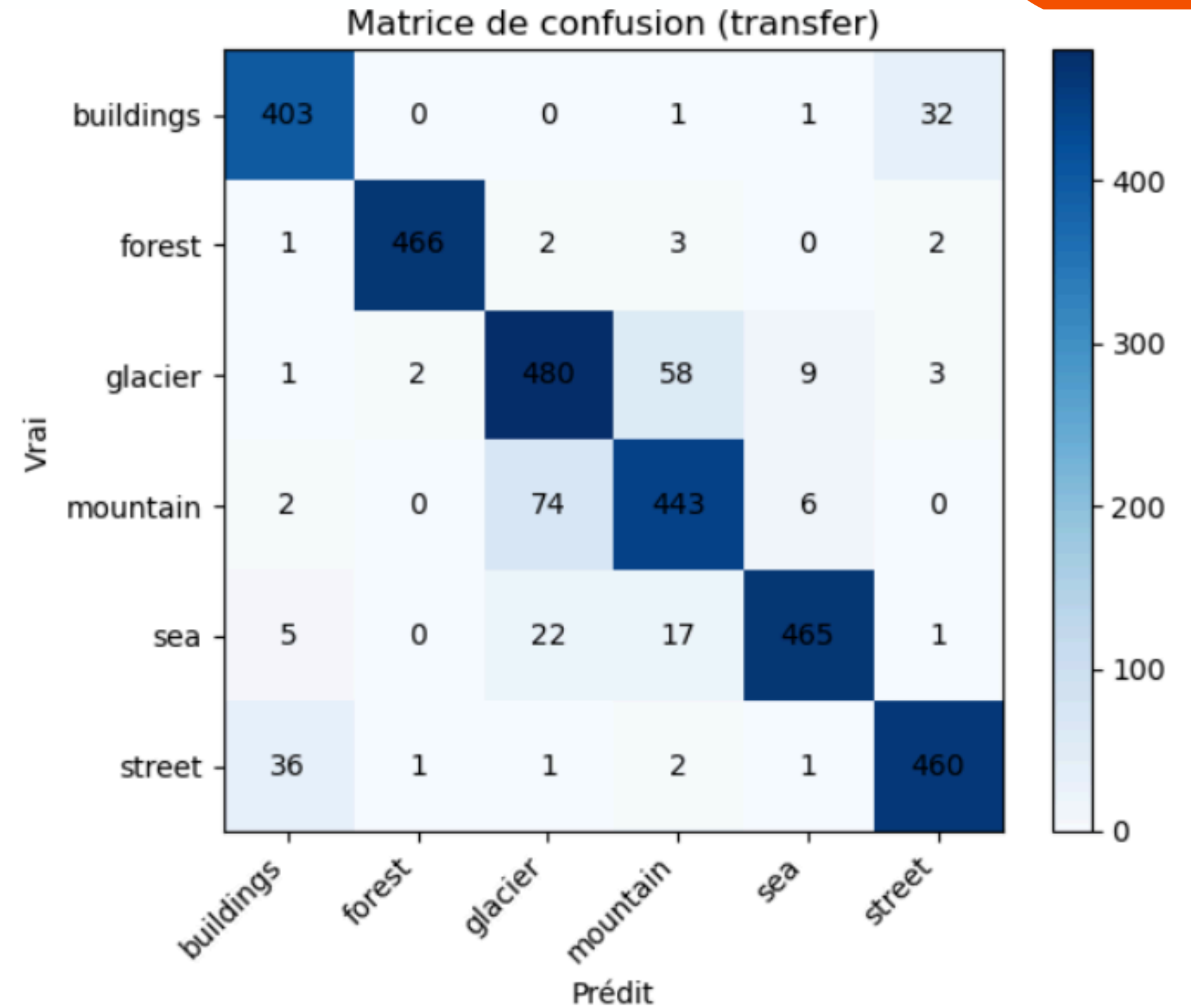
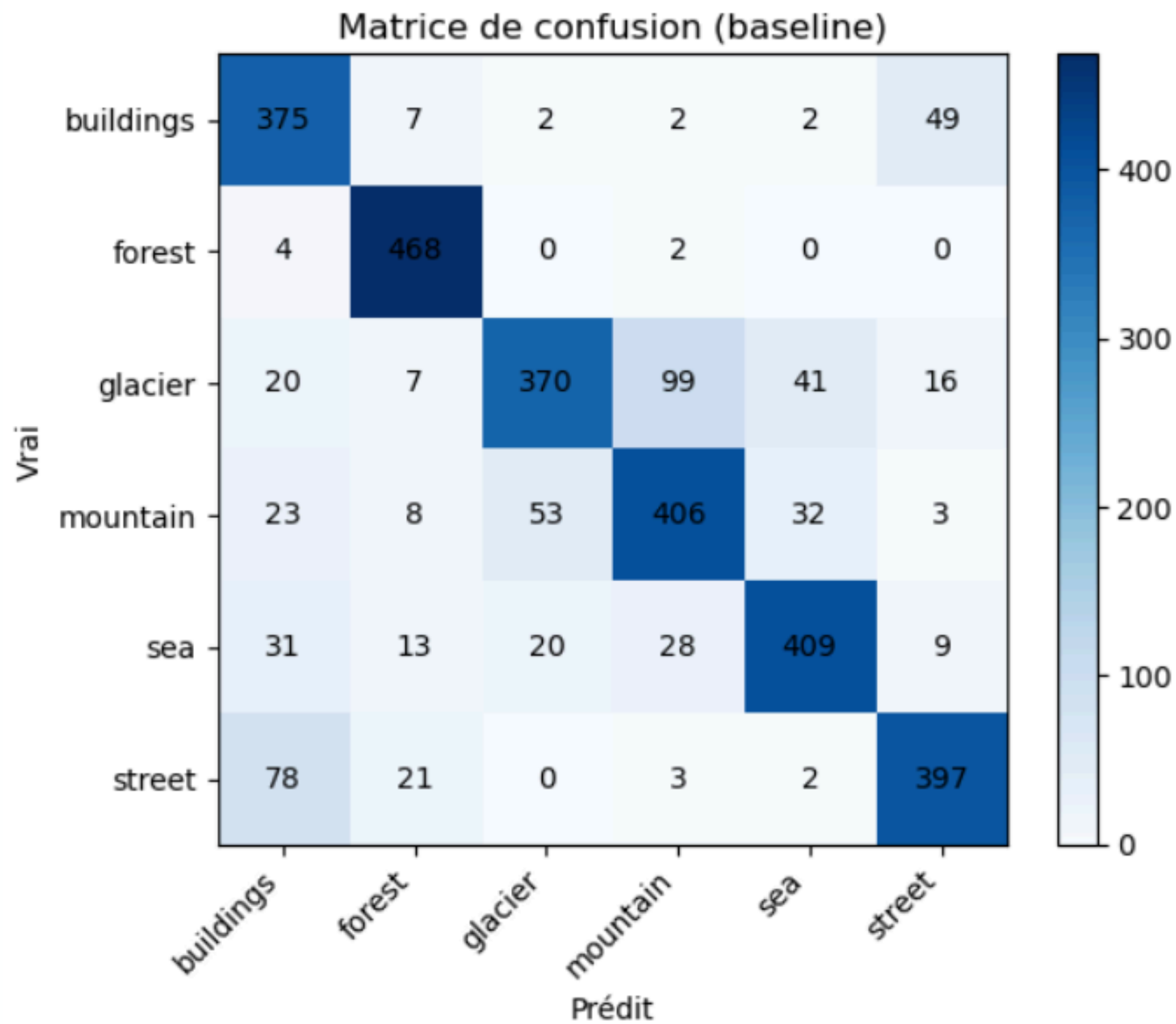
- CNN baseline : convolution + pooling + dense
- Transfer Learning : MobileNetV2 (ImageNet)
- Fine-tuning des couches profondes
- Optimiseur : Adam | Loss : Categorical Crossentropy

Résultats

Comparaison Baseline vs Transfer Learning

Classe	Precision (Baseline)	Recall (Baseline)	F1-score (Baseline)	Precision (Transfer)	Recall (Transfer)	F1-score (Transfer)
Buildings	0.71	0.86	0.77	0.90	0.92	0.91
Forest	0.89	0.99	0.94	0.99	0.98	0.99
Glacier	0.83	0.67	0.74	0.83	0.87	0.85
Mountain	0.75	0.77	0.76	0.85	0.84	0.84
Sea	0.84	0.80	0.82	0.96	0.91	0.94
Street	0.84	0.79	0.81	0.92	0.92	0.92
Accuracy	0.81	–	–	0.91	–	–
Macro avg	0.81	0.81	0.81	0.91	0.91	0.91
Weighted avg	0.81	0.81	0.81	0.91	0.91	0.91

Résultats



Application Streamlit

- Application Web pour tester le modèle
- Upload d'images multiples
- Choix du Top-k
- Visualisation des probabilités
- Interface interactive

Intel Scenes — Prédiction simple (sans Grad-CAM)

Charge un modèle .keras/.h5, envoie une image, et vois la prédiction.

Chemin du modèle

transfer_best.keras

Top-k

3

- +

i Top-k : affiche les k classes les plus probables, triées de la plus certaine à la moins certaine.

Modèle chargé.

Glisse une ou plusieurs images (JPG/PNG)...



Drag and drop files here

Limit 200MB per file • JPG, JPEG, PNG

Browse files



Image originale

Prédictions

- buildings : 95.23%
- street : 1.86%
- mountain : 1.26%

Conclusion et perspectives

- Transfer learning améliore fortement les performances
- Projet utile pour la classification d'images naturelles
- **Perspectives :**
 - Nouvelles architectures (ResNet, EfficientNet)
 - Déploiement sur le cloud (Streamlit Cloud, HuggingFace)



**MERCI POUR VOTRE
ECOUTE !**

