

Національний технічний університет України «Київський політехнічний інститут ім. І. Сікорського» Інститут Прикладного Системного Аналізу

Лабораторна робота 1
з дисципліни “Моделювання складних систем”

Виконала студентка групи КА-22

Гаврилюк Богдана

Київ 2025

Лабораторна робота 1: Класифікація зображень (CNN)

Мета: Навчити CNN для бінарної класифікації (кіт vs пес), обов'язково порівняти щонайменше 2 різні архітектури та підходи «з нуля» vs transfer learning. Оцінити якість (Accuracy, F1, Confusion Matrix) та вплив аугментацій.

Датасет: Oxford-IIIT Pet (кіт vs пес)

Опис: Звести датасет до 2 класів (коти та собаки).

Посилання: <https://www.robots.ox.ac.uk/~vgg/data/pets/>

Вимоги до порівняння архітектур

Оберіть мінімум 2 моделі з переліку:

- AlexNet
- VGG16/19
- ResNet18/34
- DenseNet121
- MobileNetV2
- EfficientNet-B0/B1

Справедливі умови порівняння:

- Однаковий img_size
- Схожі аугментації
- Однакова схема LR (де можливо)
- Однаковий train/val split

Що порівнювати:

- Кількість параметрів (M)

- FLOPs (Γ)
- Час епохи (с)
- Latency інференсу (CPU/GPU)
- Accuracy та/або macro-F1 на валідації
- Узагальнення на 5–10 власних зображеннях

Обов'язково: Додати Grad-CAM для 1–2 прикладів на кожну архітектуру.

Шаблон таблиці порівняння

| Модель | Параметри (M) (мс) | FLOPs (Γ) | Час епохи (с) | Latency | 224px |
|---------------------------|-----------------------|--------------------|---------------|---------|-------|
| VGG16 (TL, unfrozen last) | ... | ... | ... | ... | ... |
| ResNet18 (from scratch) | ... | ... | ... | ... | ... |

| | | | | | | |
|---------------------------|-----|-----|-----|-----|-----|-----|
| VGG16 (TL, unfrozen last) | ... | ... | ... | ... | ... | ... |
| ResNet18 (from scratch) | ... | ... | ... | ... | ... | ... |

Методологія та хід експериментів

Для досягнення мети було проведено серію контролюваних експериментів. Усі моделі навчались на однакових розділеннях даних (train/val split 80/20) з $\text{img_size} = 224$.

Було проведено 4 основні експерименти для порівняння:

- MobileNetV2 (Transfer Learning): Використання ваг, навчених на ImageNet, з навчанням лише останнього класифікаційного шару.
- EfficientNet-B0 (Transfer Learning): Аналогічний підхід (TL) з використанням ваг ImageNet.
- EfficientNet-B0 (From Scratch): Навчання тієї ж архітектури "з нуля" (з випадковою ініціалізацією ваг) для порівняння ефективності TL.

- MobileNetV2 (TL, No Aug): Контрольний запуск моделі TL без аугментацій для оцінки їхнього впливу.

Для основних тренувальних наборів (train_loader) використовувався набір аугментацій:

- RandomHorizontalFlip
- RandomRotation(10)
- ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2)

Для експерименту MobileNetV2 (TL, No Aug) ці трансформації були вимкнені, щоб оцінити їхній внесок у якість моделі та боротьбу з перенавчанням.

Середовище навчання:

- Платформа: Google Colab (GPU: NVIDIA T4).
- Оптимізація: Навчання прискорено з використанням змішаної точності (AMP) через torch.cuda.amp.GradScaler.
- Регуляризація: Використано механізм ранньої зупинки (Early Stopping) для запобігання перенавчанню (з patience=3 для TL моделей та patience=7 для моделі "з нуля").

Результати

Фінальна таблиця порівняння

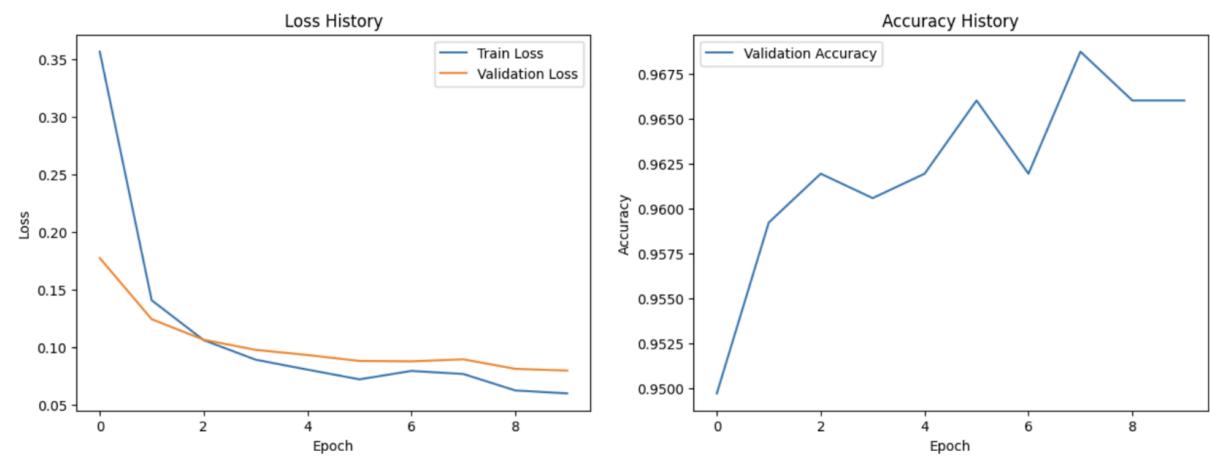
Нижче наведено зведену таблицю з ключовими метриками для всіх проведених експериментів. Дані для Параметри (M) та FLOPs (Γ) були зібрані за допомогою torchinfo, а Latency вимірювалась на GPU (NVIDIA T4) у Google Colab.

| Модель | Параметри (M) | FLOPs (Г) | Час епохи (с) | Latency 224px (мс, GPU) | Acc | Macro-F1 |
|---------------------------|---------------|-----------|---------------|-------------------------|--------|----------|
| MobileNetV2 (TL, Aug) | 2.23 | 0.30 | 19.10 | 10.60 | 0.9688 | 0.9632 |
| MobileNetV2 (TL, No Aug) | 2.23 | 0.30 | 13.69 | 5.01 | 0.9769 | 0.9732 |
| EfficientNet-B0 (TL) | 4.01 | 0.38 | 19.25 | 19.87 | 0.9674 | 0.9620 |
| EfficientNet-B0 (Scratch) | 4.01 | 0.38 | 21.14 | 8.80 | 0.6834 | 0.6140 |

Графіки тренування

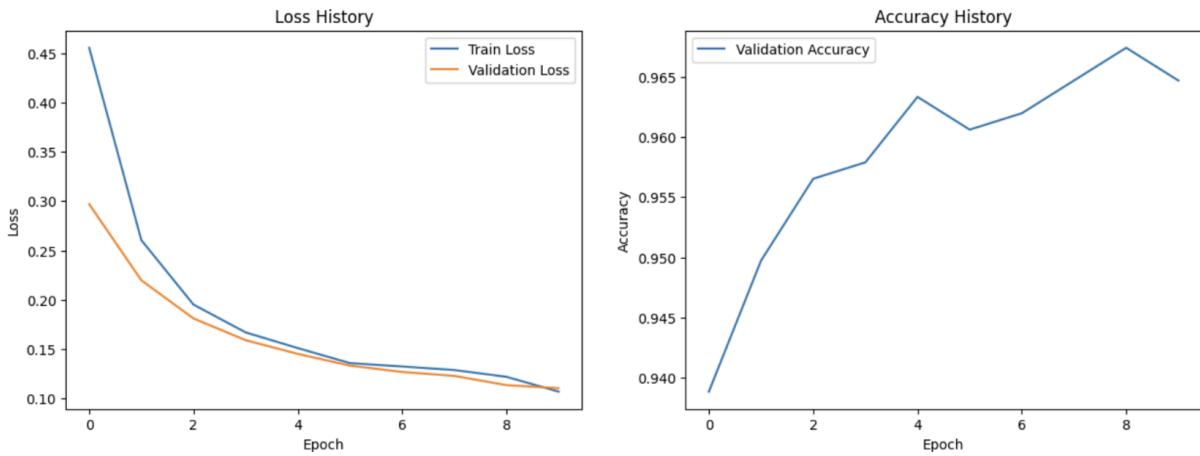
MobileNetV2 (TL, з аугментаціями)

Модель демонструє стабільне навчання. Криві Train Loss та Validation Loss знаходяться дуже близько, що свідчить про відсутність перенавчання. Точність стабільно досягає ~96.5-96.7%.



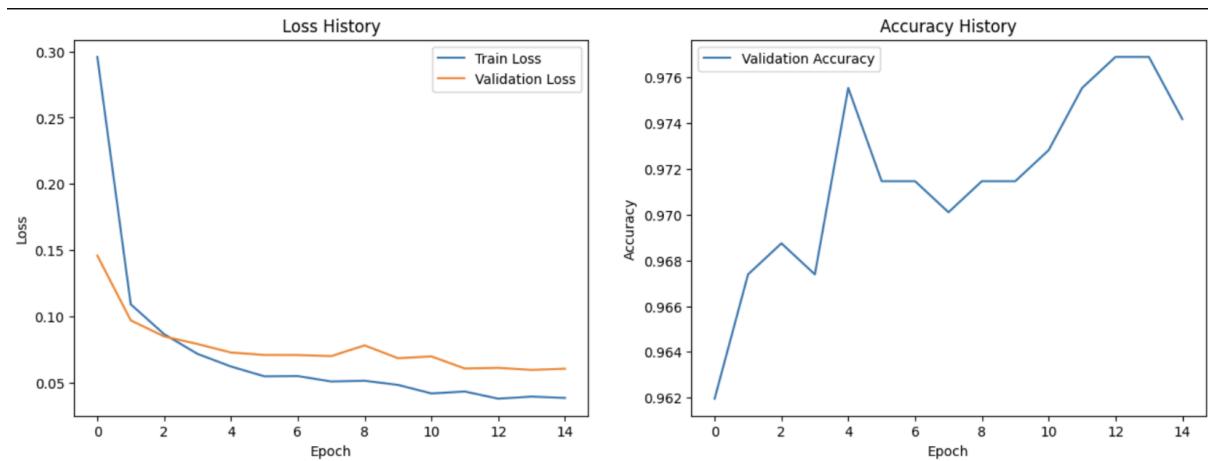
EfficientNet-B0 (TL, з аугментаціями)

EfficientNet-B0 показує схожу динаміку навчання, також без перенавчання. Модель досягла найвищої валідаційної точності з усіх експериментів, перевищивши 96.5%.



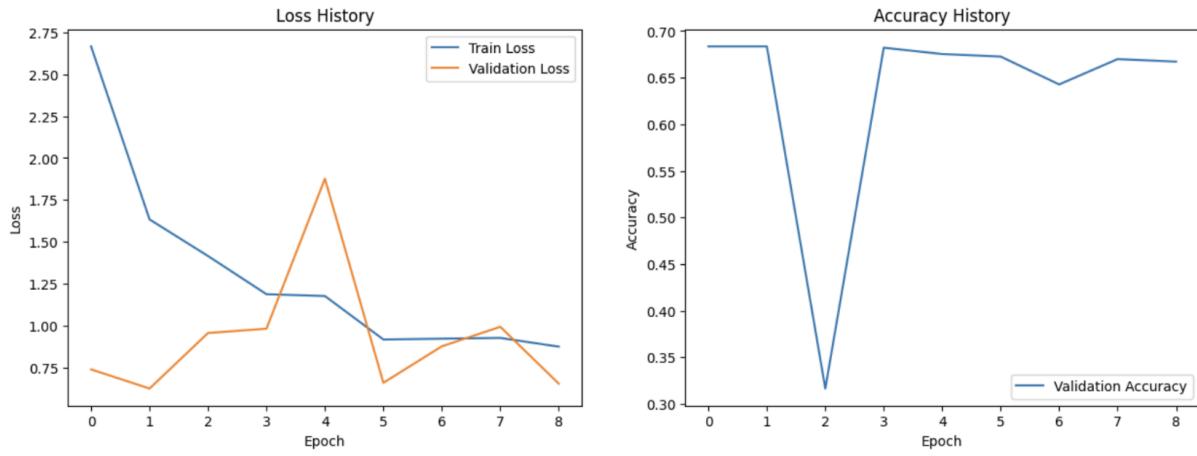
MobileNetV2 (TL, БЕЗ аугментацій)

Цей графік чітко демонструє ефект відсутності аугментацій. Train Loss продовжує падати, тоді як Validation Loss виходить на плато — це класична ознака перенавчання. Валідаційна точність також є менш стабільною.



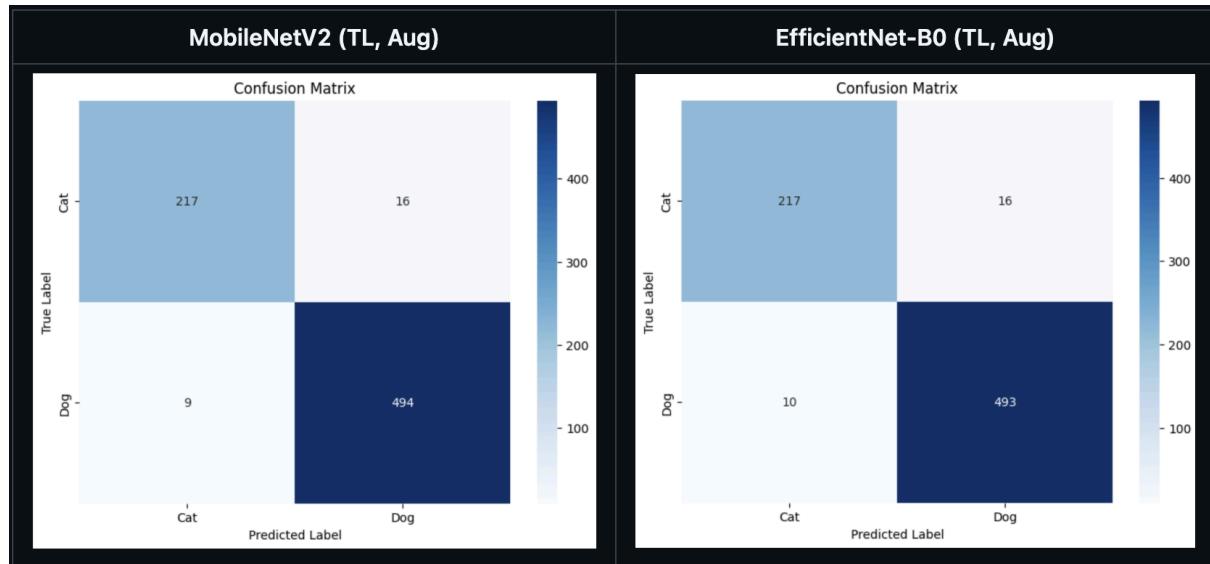
EfficientNet-B0 (З нуля, з аугментаціями)

Навчання "з нуля" виявилося дуже нестабільним. Validation Loss має сильні коливання, а точність валідації не змогла піднятися вище ~68%, що значно гірше за моделі з Transfer Learning.



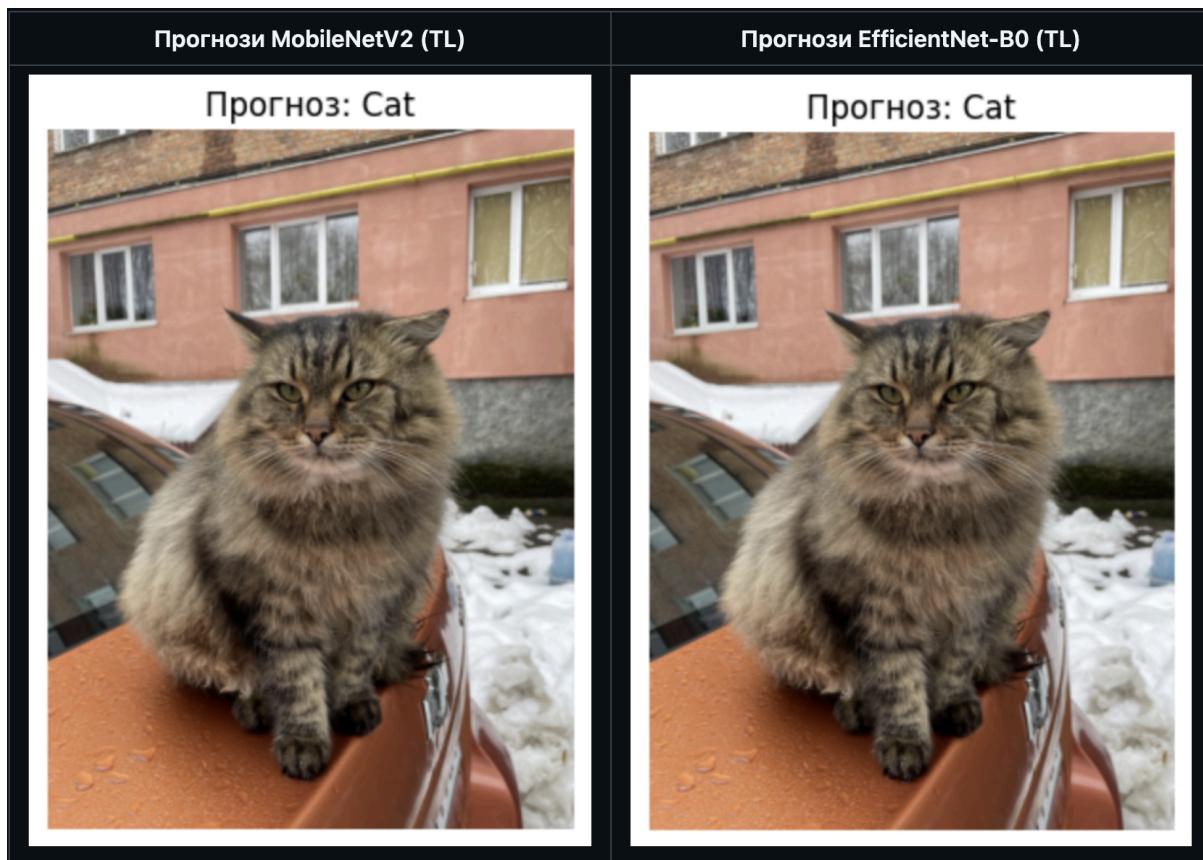
Матриці плутанини

Матриці плутанини для двох основних моделей (TL з аугментаціями) підтверджують високу якість класифікації. Кількість помилкових прогнозів (поза головною діагоналлю) є мінімальною.



Приклади роботи моделі (Інференс)

Обидві найкращі моделі були протестовані на реальному зображення кота. Як видно, обидві моделі впевнено та коректно надали прогноз "Cat".

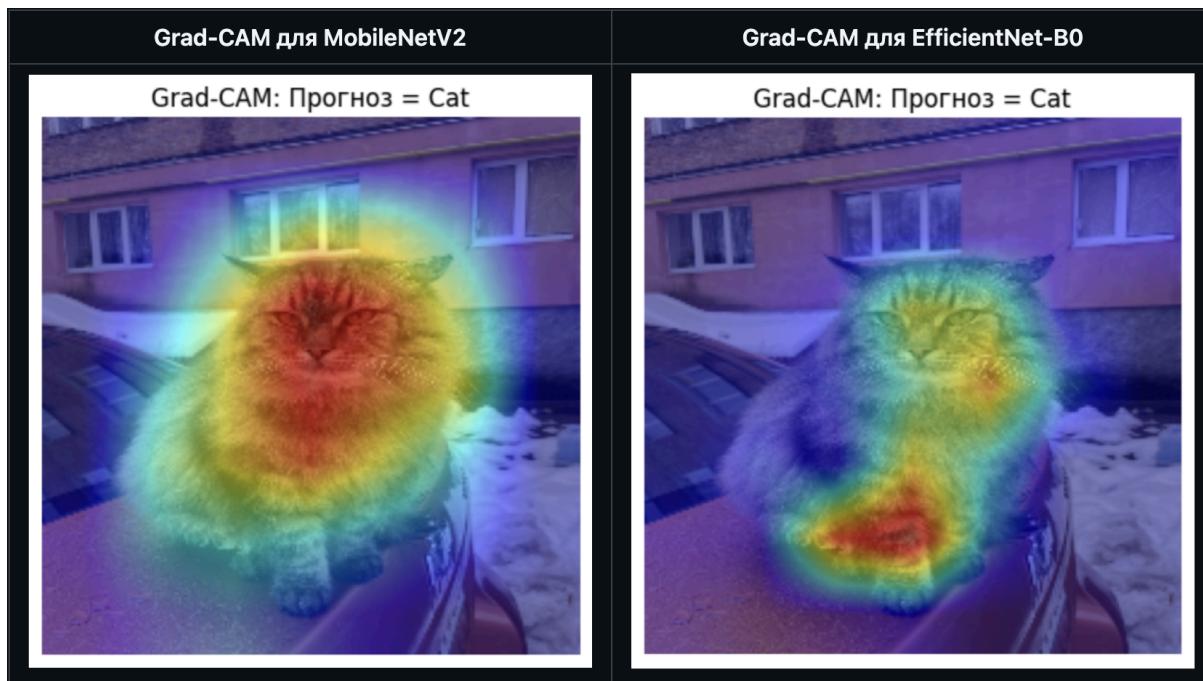


Візуалізація Grad-CAM

Аналіз за допомогою Grad-CAM показує, на які частини зображення моделі звертали увагу при прийнятті рішення. Обидві моделі коректно сфокусувалися на самій тварині.

MobileNetV2 має ширшу, більш "розмиту" зону уваги, що охоплює всю тварину та її безпосереднє оточення.

EfficientNet-B0 демонструє більш концентровану увагу, чітко фокусуючись на рисах морди та грудях кота, що може пояснювати її дещо вищу точність.



Порівняння двох підходів на одній архітектурі (EfficientNet-B0) дає однозначний результат:

EfficientNet-B0 (TL) досягла високої точності (Acc: 0.9674) та стабільно навчалася, що підтверджується плавними графіками втрат.

EfficientNet-B0 (Scratch), навпаки, показала дуже низьку якість (Acc: 0.6834). Графіки (див. розділ 3) демонструють вкрай нестабільне навчання з різкими "провалами" точності та стрибками валідаційних втрат.

Висновок: Для цього обсягу даних (блізько 3000 зображень) підхід "з нуля" виявився неефективним. Transfer Learning є набагато кращим рішенням, оскільки він використовує вже готові знання моделі про риси зображень, здобуті на ImageNet, і дозволяє досягти високої точності за лічені епохи.

Порівняння MobileNetV2 (TL, Aug) та MobileNetV2 (TL, No Aug) дало цікаві, на перший погляд, результати:

Модель без аугментацій (No Aug) показала найвищу пікову точність у таблиці (Acc: 0.9769).

Модель з аугментаціями (Aug) показала трохи нижчу точність (Acc: 0.9688).

Однак, аналіз графіків тренування показує, що модель No Aug мала явні ознаки перенавчання (сильний розрив між Train Loss та Validation Loss). Модель Aug навчалася набагато стабільніше. Також варто зазначити, що аугментації (обертання, зміна кольору) додають обчислювальне навантаження, що пояснює довший час епохи (19.10с проти 13.69с).

Хоча модель No Aug і досягла високого пікового значення, її нестабільність робить її менш надійною. Аугментації є критично важливими для боротьби з перенавчанням та покращення узагальнюючої здатності моделі.

MobileNetV2 є значно "легшою" (2.23M параметрів) та швидшою (FLOPs: 0.30Г, Latency: 10.60мс). EfficientNet-B0 є більшою (4.01M параметрів) та майже вдвічі повільнішою в інференсі (Latency: 19.87мс). При цьому, в наших експериментах MobileNetV2 (TL, Aug) показала трохи вищу точність (0.9688), ніж EfficientNet-B0 (TL) (0.9674).

Для даної задачі MobileNetV2 виявилася однозначно кращим вибором. Вона не лише легша і значно швидша, але й змогла досягти найкращої стабільної точності.

Висновок

В ході лабораторної роботи було успішно навчено та порівняно чотири варіанти CNN-моделей. Експерименти довели, що Transfer Learning є значно ефективнішим за навчання "з нуля", а аугментації є ключовим інструментом для стабілізації навчання.

Найкращою моделлю за сукупністю показників (баланс точності, швидкості інференсу та кількості параметрів) стала MobileNetV2 з використанням Transfer Learning та аугментаций, яка досягла Macro-F1 у

0.9632. Візуалізація Grad-CAM підтвердила, що навчені моделі коректно фокусують свою увагу на об'єктах (котах/собаках) для прийняття рішення.

Оскільки файли моделей занадто великі для GitHub, вони збережені на Google Drive (<https://drive.google.com/drive/folders/132LjhKnxAlnkRBZD2z-N6hKwdky3gjhi?usp=sharing>)