**Проєкт “Розпізнавання ігор по зображенню геймплею”**

**Звіт**

Студент: Бондаренко Віталій Віталійович

**Підготовка**

Основною метою є створення моделі, яка буде розпізнавати гру, до якої належить зображення. Нижче наведені приклади зображень, на яких буде навчатися модель:

Зображення, що містить текст, знімок екрана, Комп’ютерна гра, Програмне забезпечення для відеоігор

Автоматично згенерований опис

Зображення, що містить Комп’ютерна гра, карта, текст, знімок екрана

Автоматично згенерований опис

Зображення, що містить транспортний засіб, знімок екрана, автомобіль, просто неба

Автоматично згенерований опис

Буду створювати модель, на основі вже навченої **ResNet18**, налаштованої для класифікації на класи, що містяться в моєму наборі даних.

Основні бібліотеки, що використовуються в проєкті, включають **PyTorch**, **Torchvision**, **NumPy**, **Matplotlib** та **Splitfolders**. Кожна з цих бібліотек має важливу роль у побудові, тренуванні та оцінці нейронної мережі.

Імпортуємо їх:

Зображення, що містить текст, Шрифт, програмне забезпечення, знімок екрана

Автоматично згенерований опис

Далі за допомогою бібліотеки **splitfolders** сформую правильний датасет:



Далі підготовлюю дані для навчання та валідації моделі:

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Написав функції для візуалізації зображень із тренувального набору, що дозволяє наочно оцінити вигляд вхідних даних після застосування трансформацій та нормалізації:

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Спробуємо вивести зображення з батчу:

Зображення, що містить текст, знімок екрана, Веб-сайт, Веб-сторінка

Автоматично згенерований опис

**Навчання та визначення найкращої моделі**

Для початку написав функцію **train\_model,** яка відповідає за навчання та валідацію нейронної мережі. Функція дозволяє оцінювати продуктивність моделі на кожній епосі, зберігати найкращі ваги та коригувати швидкість навчання відповідно до планувальника (scheduler). Це ключовий елемент у процесі тренування, що забезпечує можливість контролювати втрати та точність на тренувальній і валідаційній вибірках, а також зберігати найкращу версію моделі:

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, документ

Автоматично згенерований опис

Зображення, що містить текст, Шрифт, ряд, знімок екрана

Автоматично згенерований опис

Також написав функцію **visualize\_model,** яка відповідає за візуалізацію передбачень моделі на валідаційних даних. Основне завдання цієї функції — наочно показати, які класи модель прогнозує для зображень із валідаційного набору, що дозволяє оцінити її точність та можливі помилки:

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Налаштувую та треную модель **ResNet18** для класифікації зображень, включаючи визначення критерію втрат, вибір оптимізатора та планувальника швидкості навчання:

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Результати навчання:

Зображення, що містить текст, Шрифт, знімок екрана, документ

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, документ

Автоматично згенерований опис

* Модель досягла дуже високої точності на валідаційних даних (99.10%), що свідчить про ефективне навчання без очевидного перенавчання.
* Невеликі зміни в точності та втрати в кінцевих епохах вказують на те, що навчання досягло своєї оптимальної точки. Втрати стабілізувалися, і точність зросла до майже ідеального рівня.

Отож, модель **ResNet18** була ідеально навчена для цього завдання, досягнувши високої точності з коротким часом навчання. Результати розпізнавання моделі:

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Для наступного експерименту я хотів би прибрати аугментацію даних та навчити розпізнавати зображення за допомогою вже навченої мережі **EfficientNetB0.**

Це теж глибока нейронна мережа, але від **ResNet18** вона буде відрізнятися своєю архітектурою.

Вона використовує **комплексний підхід масштабування** (Compound Scaling), який одночасно збільшує ширину, глибину та роздільну здатність зображень, щоб забезпечити максимальну продуктивність при мінімальних витратах ресурсів. Також вона використовує **мобільні згортки** (Mobile Convolutions), що робить її ефективною як для обчислень, так і для пам’яті, дозволяючи досягти високої точності з меншою кількістю параметрів, ніж у **ResNet18.**

Першим кроком підготуємо дані без аугментації:

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Відсутні операції аугментації, такі як випадкове обертання, випадкове обрізання, горизонтальне обертання або зміна кольорових характеристик зображення, які зазвичай додаються для збільшення варіативності даних і покращення узагальнення моделі.

Завантажую модель **EfficientNetB0** (оптимізатор та функцію втрат залишив попередні)**:**

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Функція тренування та візуалізації залишилися незмінними.

Навчаємо модель:



Результати навчання:

Зображення, що містить текст, Шрифт, знімок екрана, документ

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, документ

Автоматично згенерований опис

* **Точність на тренувальній вибірці** швидко зросла, досягнувши 99.29% після 10 епох. Це свідчить про те, що модель добре вивчає тренувальні дані.
* **Точність на валідаційній вибірці** також дуже висока, досягнувши максимуму 99.25% на 6-й епосі. Цей рівень точності є вражаючим, особливо враховуючи, що ми не використовували аугментацію.
* **Втрати** для тренувального та валідаційного наборів стабільно знижувалися, що свідчить про добре збалансований процес навчання.
* На кінець 10-ї епохи можна помітити невелике зростання втрат на валідаційній вибірці (0.0364) в порівнянні з мінімальною втратою (0.0251) на 6-й епосі. Це може бути ранньою ознакою **перенавчання**, коли модель починає запам’ятовувати тренувальні дані замість того, щоб покращувати узагальнення.

Результати розпізнавання моделі:

Зображення, що містить текст, знімок екрана, дизайн

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, просто неба, автомобіль

Автоматично згенерований опис

В останньому експерименті я буду використовувати модель ResNet50 з повною аугментацією та раннім призупиненням. Це вже більш глибша модель з 50-ма шарами, яка використовує більш складні залишкові блоки. Завдяки глибині й більшій кількості параметрів, ResNet50 краще виявляє складні патерни в даних, але також вимагає більше обчислювальних ресурсів і часу для тренування

Імпортуємо необхідні бібліотеки:

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Підготовка даних з повною аугментацією:

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Завантаження моделі, налаштування оптимізатора та функції втрат:

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Реалізую нову функцію тренування з раннім призупиненням:

Зображення, що містить текст, знімок екрана, Шрифт, документ

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, документ

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Навчаємо модель:



Результати навчання:

Зображення, що містить текст, знімок екрана, Шрифт, документ

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, документ

Автоматично згенерований опис

Модель досягла дуже високих результатів із максимальною точністю 99.05% на валідаційній вибірці. Це свідчить про те, що модель добре навчається і узагальнює на нові дані. Подальші експерименти можуть бути спрямовані на дрібне налаштування (наприклад, додаткові техніки аугментації або регуляризацію), але в цілому модель вже показує відмінну продуктивність.

ВИСНОВОК

В результаті дослідження маємо наступне:

 **ResNet18 (базова модель)**:

* **Точність на тренувальній вибірці**: помірна, але стабільно зростає з кожною епохою.
* **Точність на валідаційній вибірці**: досягає прийнятного рівня, але нижча, ніж у двох інших моделей.
* **Переваги**: ResNet18 є найлегшою моделлю з трьох, з меншою кількістю параметрів. Вона швидше навчається та потребує менше обчислювальних ресурсів.
* **Недоліки**: модель має дещо нижчу точність і менш здатна до розпізнавання складних патернів у даних порівняно з ResNet50 та EfficientNetB0.

 **ResNet50 (модель з обмеженою аугментацією)**:

* **Точність на тренувальній вибірці**: висока, показує стабільне зростання з кожною епохою.
* **Точність на валідаційній вибірці**: досягає 99.05%, що є дуже хорошим результатом.
* **Переваги**: ResNet50 має більшу здатність до розпізнавання складних патернів завдяки глибині (50 шарів) та використанню bottleneck блоків. Це робить її добре придатною для роботи з великими обсягами даних і складними класами.
* **Недоліки**: ResNet50 має більше параметрів, ніж ResNet18 та EfficientNetB0, і потребує більше обчислювальних ресурсів та часу для тренування. Можливе перенавчання, якщо набір даних обмежений.

 **EfficientNetB0 (модель без аугментації)**:

* **Точність на тренувальній вибірці**: швидко зростає, досягаючи 99% вже на 6-й епосі.
* **Точність на валідаційній вибірці**: досягає 99.25% на валідаційній вибірці на 6-й епосі, що є найкращим результатом серед трьох моделей.
* **Переваги**: EfficientNetB0 є найефективнішою моделлю з трьох, досягаючи високої точності при меншій кількості параметрів. Завдяки інноваційному підходу до масштабування (Compound Scaling) та оптимізованій архітектурі, вона досягає високої точності при низьких обчислювальних витратах.
* **Недоліки**: EfficientNetB0 є більш складною в налаштуванні порівняно з ResNet. Вона може вимагати тонкого підбору гіперпараметрів для досягнення найкращих результатів.

**Найкращою моделлю серед трьох є EfficientNetB0**, і ось чому:

1. **Точність**: EfficientNetB0 показала найвищу точність на валідаційній вибірці (99.25%), що є найкращим результатом серед трьох моделей.
2. **Ефективність**: EfficientNetB0 має менше параметрів, ніж ResNet50, але досягає аналогічної або кращої точності. Це робить її більш ефективною з точки зору обчислювальних ресурсів.
3. **Здатність до узагальнення**: EfficientNetB0 демонструє високу здатність до узагальнення, навіть без використання повної аугментації, що свідчить про її здатність працювати з різними типами даних.