

Моделі нейронних мереж для класифікації пошкоджень на склі автомобіля

Виконав: Славінський
Всеволод

Науковий керівник:
Недашківська Надія
Іванівна

Постановка задачі

Завдання полягає в тому щоб створити просте і гнучке рішення на основі веб і мобільного додатків, яке б дозволяло дуже просто створювати підтвердження для страхових компаній

Важливість роботи полягає в створенні моделі нейронної мережі, яка допоможе з класифікацією пошкоджень на склі автомобіля. Модель, отримана в результаті, значно полегшить роботу страховим компаніям у визначенні пошкоджень, а також зробить систему, яка не буде залежати від людського фактору, що позитивно вплине на страхових компаніях, а також на посередників

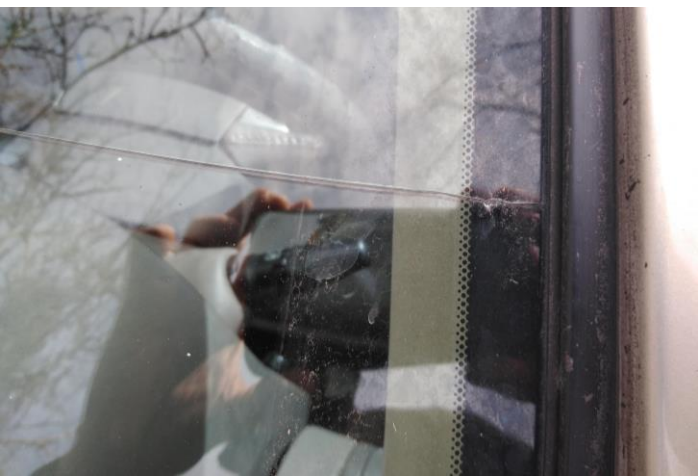
Опис системи: необхідно розробити систему, яка отримувала б фотографію на вхід, після цього користувач міг би виділити зону, на якій є пошкодження, а система відповіла, до якої з категорій це пошкодження належить.

Актуальність роботи

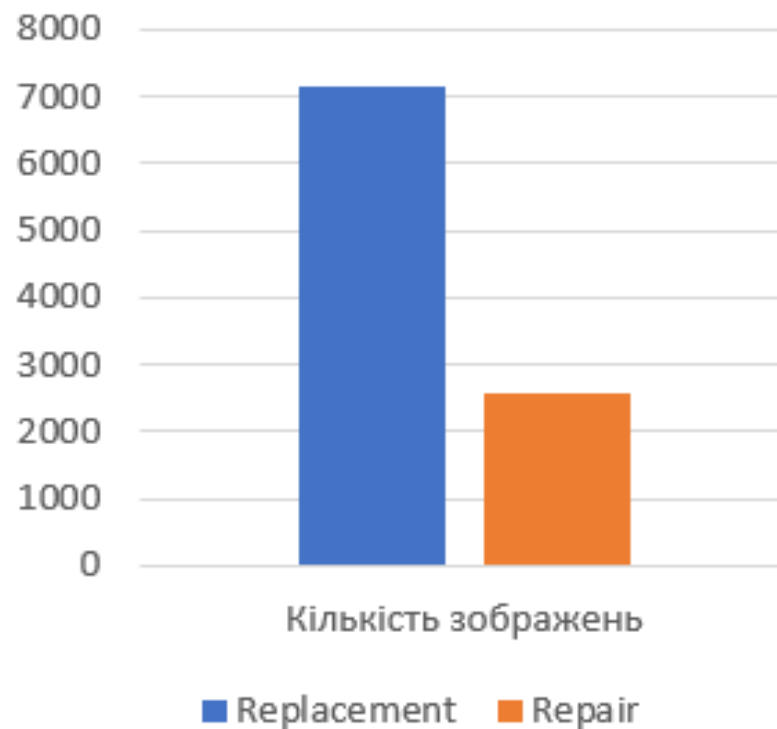
Задача є актуальною, тому що страхова сфера в США є досить складною і з великою кількістю різноманітних нюансів. Також, страхові компанії досить закриті і користуються застарілим програмним забезпеченням або паперовим видом оформлення страхових випадків.

Предмет та об'єкт досліджень

- Об'єктом дослідження є аналіз зображень з пошкодженнями на склі автомобіля, які були отримані за допомогою реальних користувачів вже працюючого додатка, який було написано для однієї з компаній-клієнтів, а також зображення отриманні в результаті алгоритмів аргументації.
- Предметом дослідження визначено згорткові нейронні мережі для класифікації зображень .



Розподіл зображень за категоріями



Початкові дані

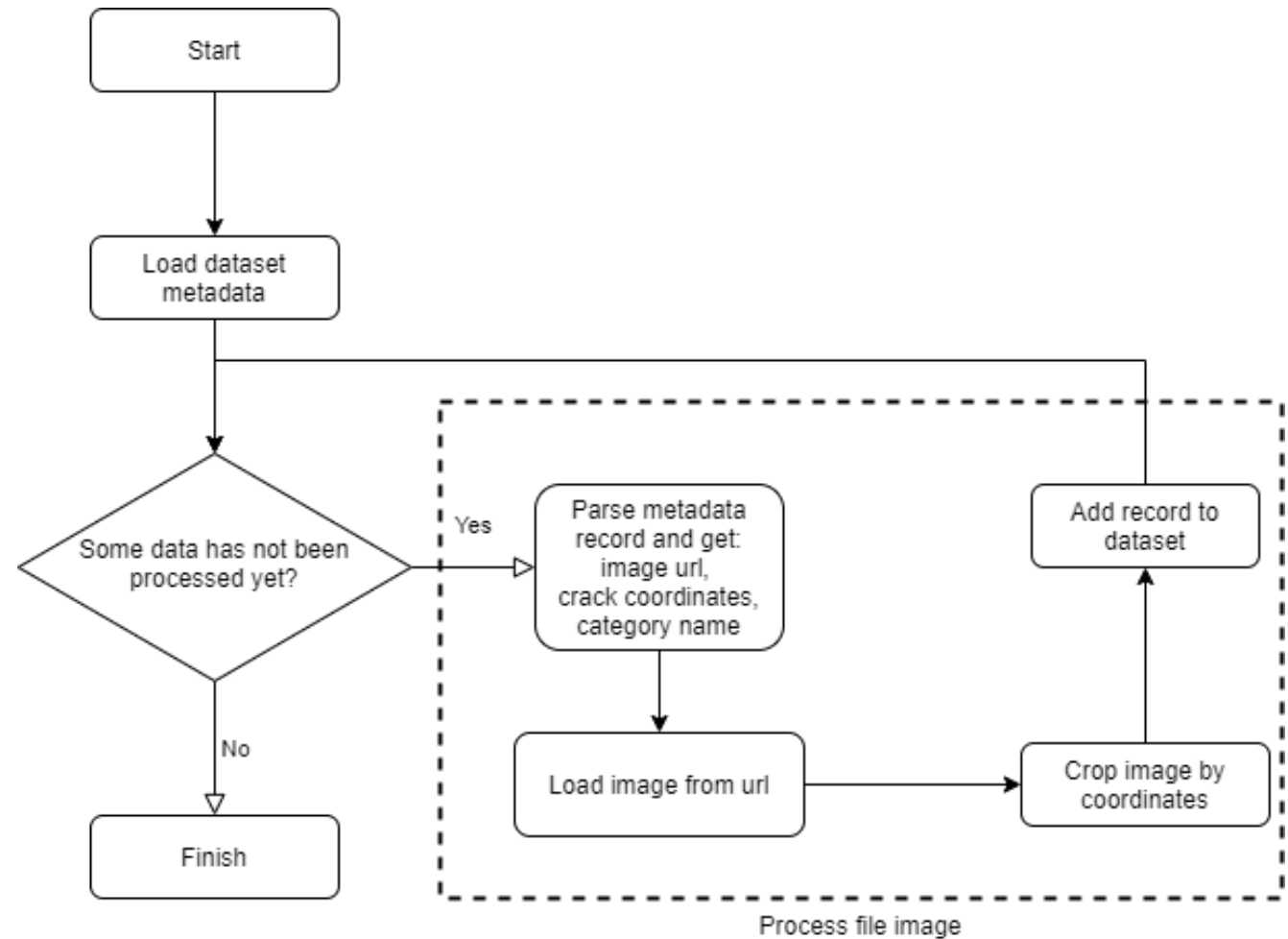
Набір даних був сформований реальними користувачами додатка, розробленого працівниками компанії WebLegends у рамках контракту з компанією, що займається випадками автострахування, та попередньо відредагований мною.

Функції для обробки даних

- `def load_image(path)` – завантажує зображення для подальшої обробки
- `def show_image(img)` – виводить на екран задане зображення
- `def crop_polygon(img, polygon)` – вирізає прямокутник на зображенні по координатах
- `def process_file_image(label_path)` – головний цикл, в якому відбувається парсинг та обробка даних

Схема процесінгу даних

- Вхідні дані – необроблені картинки та файл dataset_metadata, де зберігаються відомості про кожне з зображень
- Вихідні дані – оброблений датасет, готовий до роботи



Вихідні дані (датасет, ГОТОВИЙ ДО роботи)

Replacement



Repair



Replacement



Replacement



Repair



Replacement



Repair



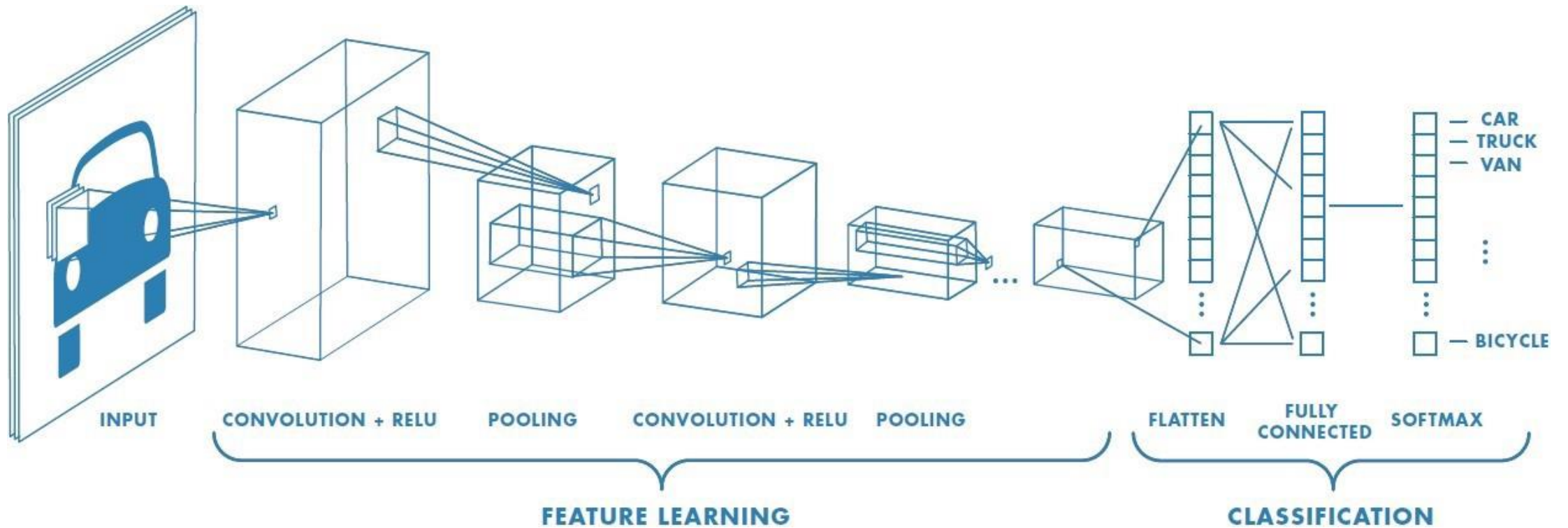
Repair



Replacement



Згорткові нейронні мережі



Вимоги до системи

Необхідно, щоб до системи можна було звернутися з будь-якого пристрою.

Система не повинна вимагати завантаження будь яких додаткових модулів або бібліотек для своєї роботи.

Різні компоненти не повинні залежати один від одного, щоб можна було розробляти їх паралельно.

Система повинна мати зрозумілий користувальницький інтерфейс, оскільки основна аудиторія – це люди, які не дуже гарно працюють з комп'ютерами

Система повинна мати зрозумілий програмний інтерфейс, щоб у майбутньому її можна було легко інтегрувати в іншу систему.

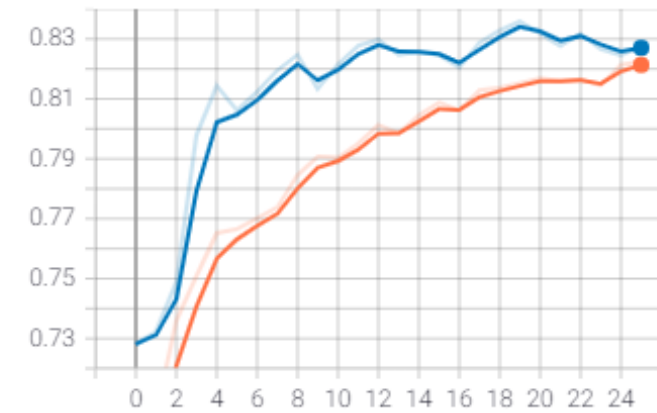
Архітектура системи

Кожне з вимог досить важливе і ми не можемо ігнорувати їх. Отже, щоб система задовольняла усім вимогам, було вирішено зробити веб додаток, який використовує попередньо навчені моделі згорткових нейронних мереж

Особливості такої моделі полягають в тому, що користувач відправляє певний запит на сервер, де той системно обробляється і кінцевий результат відсилається клієнту.

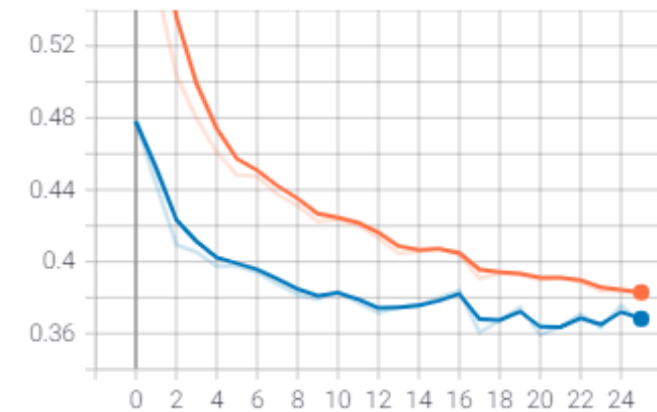
epoch_accuracy

epoch_accuracy
tag: epoch_accuracy



epoch_loss

epoch_loss
tag: epoch_loss

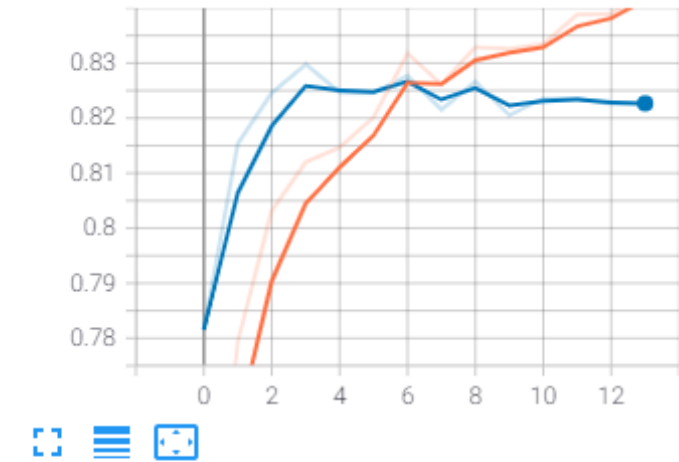


Inception v3

	Train	Validation	Test
Accuracy	0.8225	0.8277	0.8441
Loss	0.3822	0.3662	0.3416

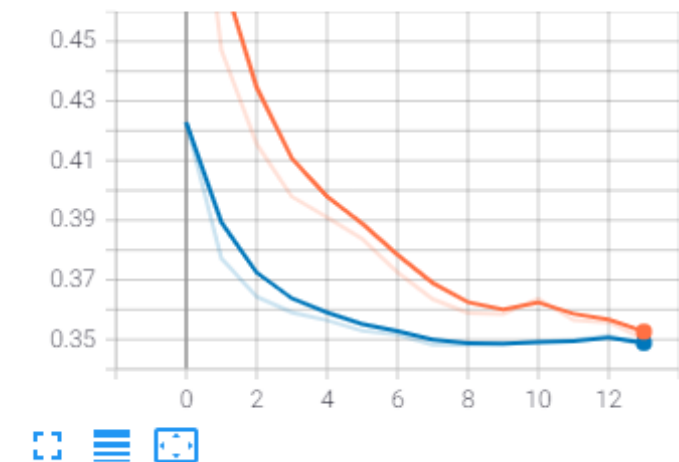
epoch_accuracy

epoch_accuracy
tag: epoch_accuracy



epoch_loss

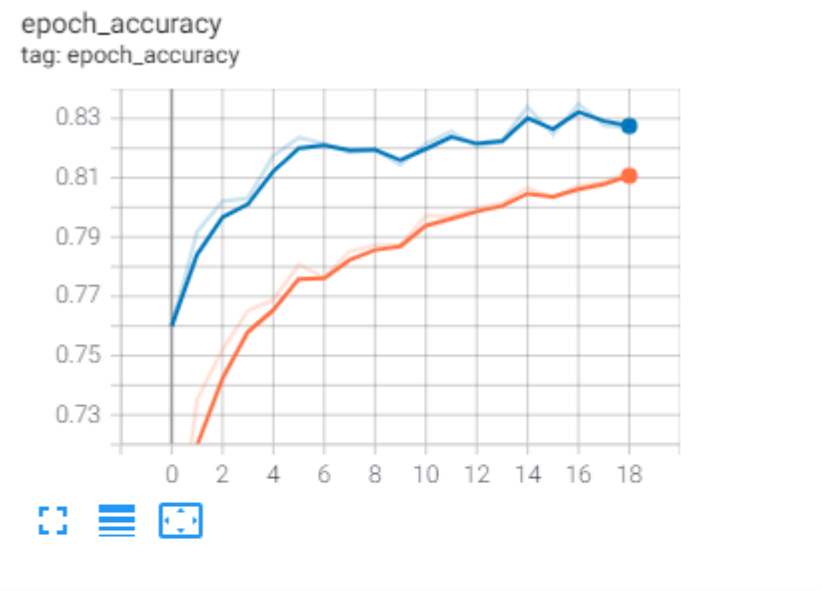
epoch_loss
tag: epoch_loss



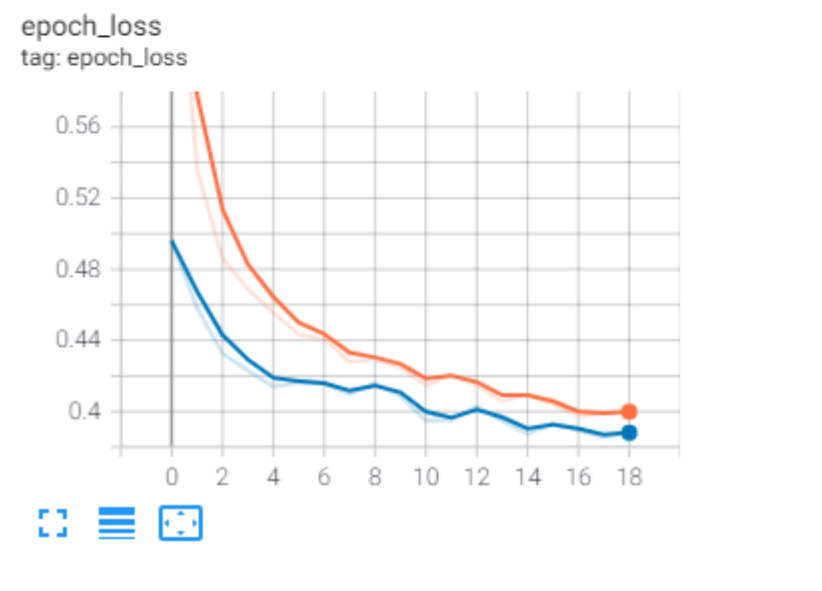
EfficientNet

	Train	Validation	Test
Accuracy	0.8434	0.8226	0.8584
Loss	0.3503	0.8434	0.3357

epoch_accuracy



epoch_loss

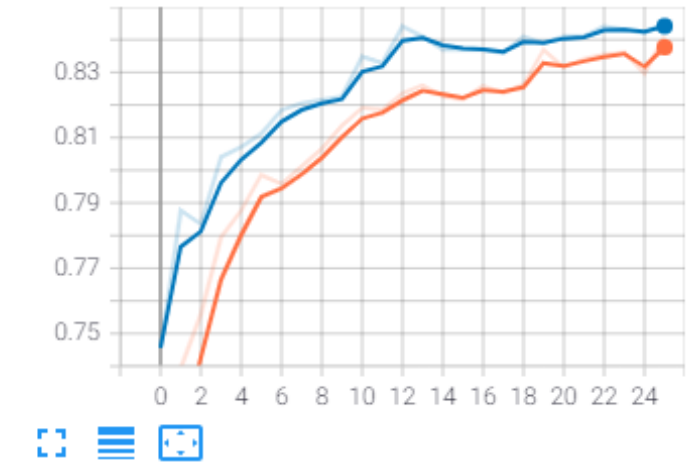


MobileNet

	Train	Validation	Test
Accuracy	0.8120	0.3886	0.8482
Loss	0.4002	0.8267	0.3491

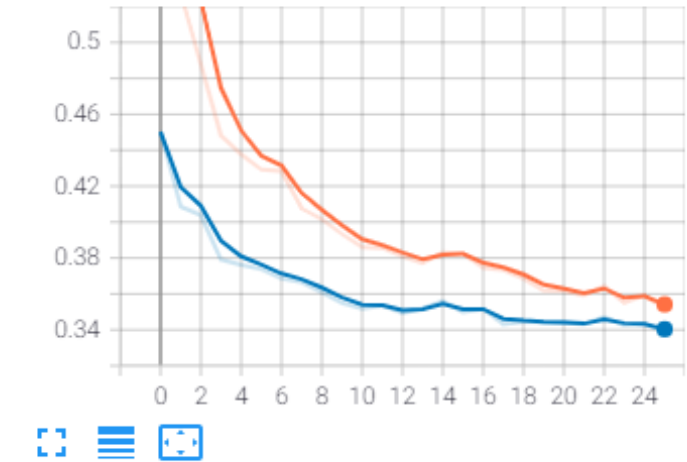
epoch_accuracy

epoch_accuracy
tag: epoch_accuracy



epoch_loss

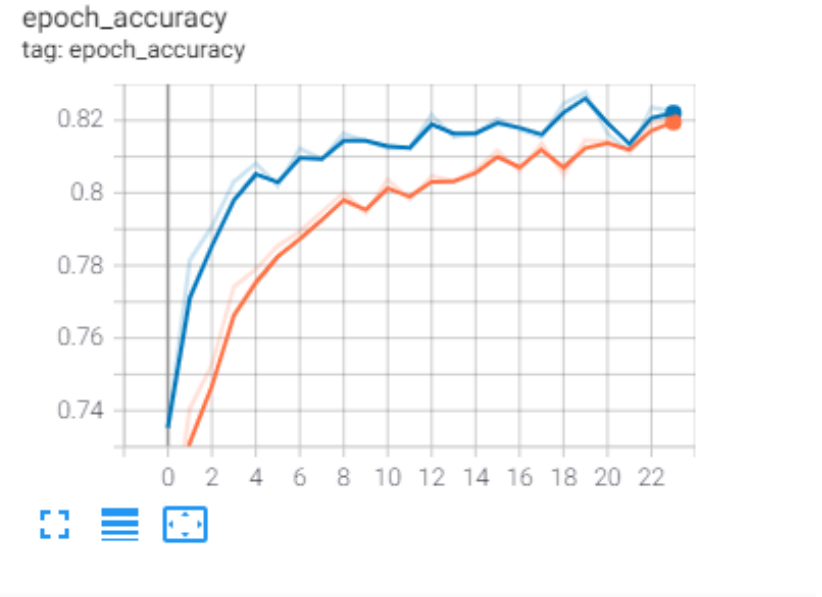
epoch_loss
tag: epoch_loss



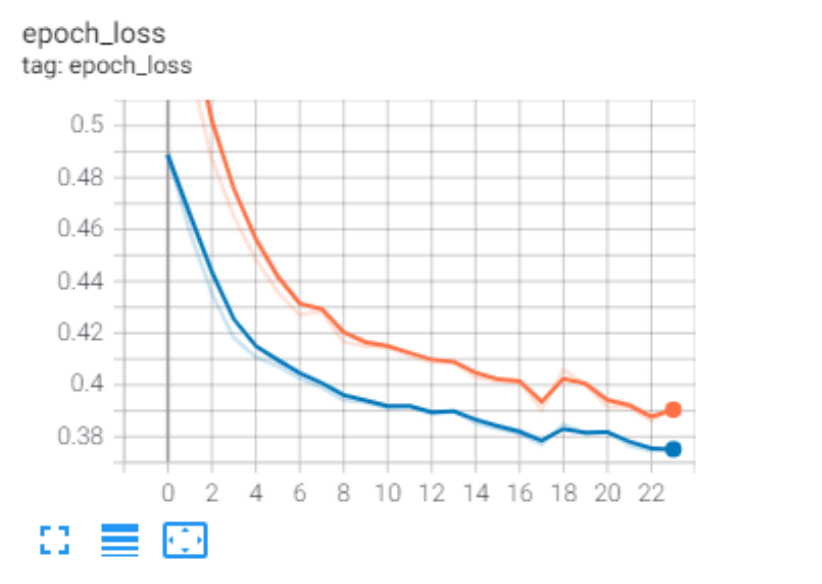
ResNet

	Train	Validation	Test
Accuracy	0.8410	0.8451	0.8471
Loss	0.3516	0.3387	0.3306

epoch_accuracy



epoch_loss



NasNet




	Train	Validation	Test
Accuracy	0.8205	0.8226	0.8461
Loss	0.3915	0.3750	0.3537


Порівняння архітектур

Архітектура	Test accuracy	Test loss
Inception	0.8441	0.3416
EfficientNet	0.8584	0.3357
ResNet	0.8471	0.3306
MobileNet	0.8482	0.3491
NasNet	0.8461	0.3537
Краща	EfficientNet	ResNet


Веб додаток

 Auto cracks classification

1. Please choose photo, that contains some cracks on auto window

 Image input


2. Please select the damaged area in the photo.


no content yet

preview area

3. Please choose model, that you want to use for classification

Inception_v3 (pretrained)

☐ Smart decision 

CLASSIFY!

Результати роботи додатку

Auto cracks classification

1. Please choose photo, that contains some cracks on auto window

Image input
20192801-...2D800.jpeg

2. Please select the damaged area in the photo.



3. Please choose model, that you want to use for classification

EfficientNet (pretrained)

☐ Smart decision ?

CLASSIFY!

Result: **Repair**

Висновки

Виконано попередній аналіз та обробку вхідних даних для подальшого використання у роботі.

Побудовано моделі нейронних мереж для класифікації пошкоджень на склі авто.

Виконано порівняльний аналіз найбільш популярних архітектур нейронних мереж.

Вивчено теоретичні основи згорткових нейронних мереж.

Зроблено веб додаток для зручної роботи з моделями нейронних мереж

Обрані найкращі моделі за метриками якості для вирішення практичної задачі класифікації пошкоджень скла.

Перспективи подальших досліджень

Спробувати нові, більш складні архітектури мереж

Додати розпізнавання знаходження пошкоджень

Навчити мережу на більшій кількості категорій

Додати нові підкатегорії

Опублікувати веб додаток у мережі

Дослідити роботу автокодувальників

Дякую за увагу!

