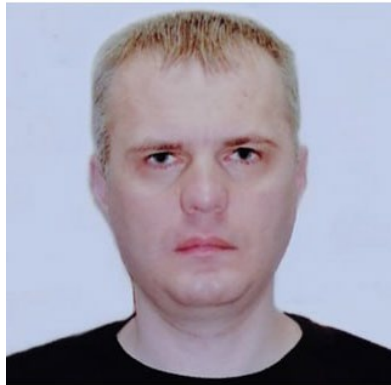


Abstract geometric lines in the top left corner, consisting of several overlapping, tilted rectangles and polygons in a light beige color.

VerifyLens

Розпізнавання правдивості зображень за допомогою AI

НАША КОМАНДА



Валерій Третяков
Team Lead



Сергій Підкопай
Developer



Василь Глущенко
Developer



Микола Присташ
Scrum Master



ПОРТРЕТ КОРИСТУВАЧА

Діяльність

Маркетолог, який прагне вдосконалити взаємодію з користувачами та якість контенту у соціальних мережах. Він шукає ефективний спосіб перевірки правдивості категорій зображень та покращення процесу взаємодії між користувачами та платформою

Потреби

Автоматизована перевірка даних: розробка ефективних способів підтвердження правдивості категорій, вказаних користувачами під час публікацій у соціальних мережах.

Точність та надійність моделі: надійна згортова нейронна мережа, яка буде ефективно визначати, чи відповідають обрані категорії зображенням. Точність є ключовим аспектом для забезпечення якісної перевірки.

Користувацький зручний інтерфейс: зручний інтерфейс для взаємодії з моделлю.



МЕТА

Розробка і реалізація надійної згорткової нейронної мережі, яка буде ефективно визначати, чи відповідає зображення користувача одній з обраних категорій, та створення зручного інтерфейсу для взаємодії з цією мережею

ЗАВАНТАЖЕННЯ ТА ДОСЛІДЖЕННЯ ДАТАСЕТУ

Підготувати датасет, розділяючи його на тренувальний, валідаційний та тестовий набори даних. Використати 80% даних для тренування, 10% для валідації та 10% для тестування. Виконати ресайз зображень та нормалізувати значення пікселів

РОЗРОБКА ТА ОЦІНКА МОДЕЛІ

Вибрати архітектуру, з кількома згортковими та пулінговими шарами. Встановити відповідні функції втрат, оцінити модель на валідаційних даних. При необхідності донавчити модель на комбінованих тренувальних та валідаційних даних

ТЕСТУВАННЯ МОДЕЛІ

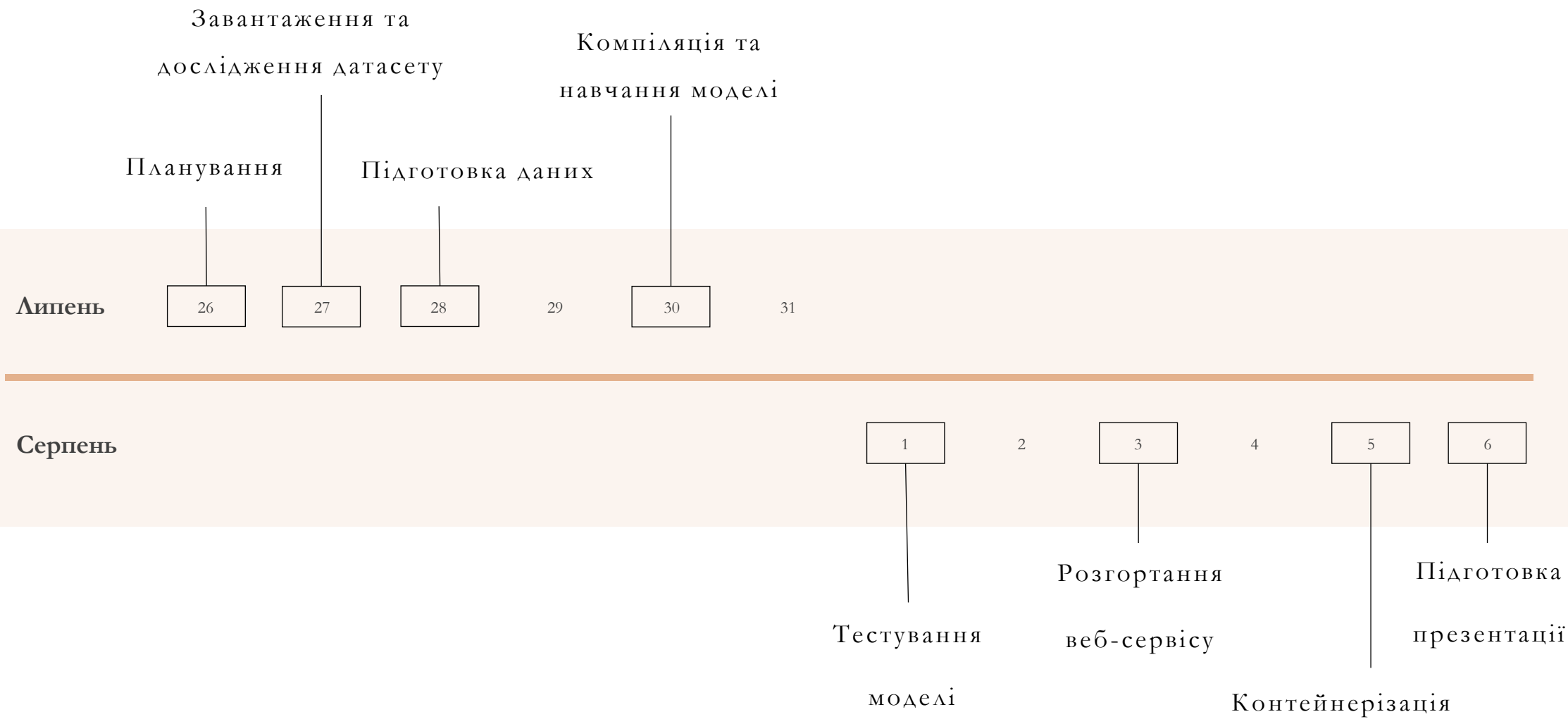
Після завершення навчання перевірити модель на тестових даних, які раніше не використовувались для навчання чи валідації.

РОЗГОРТАННЯ ВЕБ-СЕРВІСУ

Розгорнути модель на веб-сервері, забезпечуючи API для передачі зображень та отримання передбачень. Перевірити працездатність веб-сервісу, надсилаючи тестові зображення та перевіряючи, чи повертаються відповідні класифікаційні результати.

ЗАВДАННЯ

ДВОТИЖНЕВИЙ ПЛАН РОЗРОБКИ

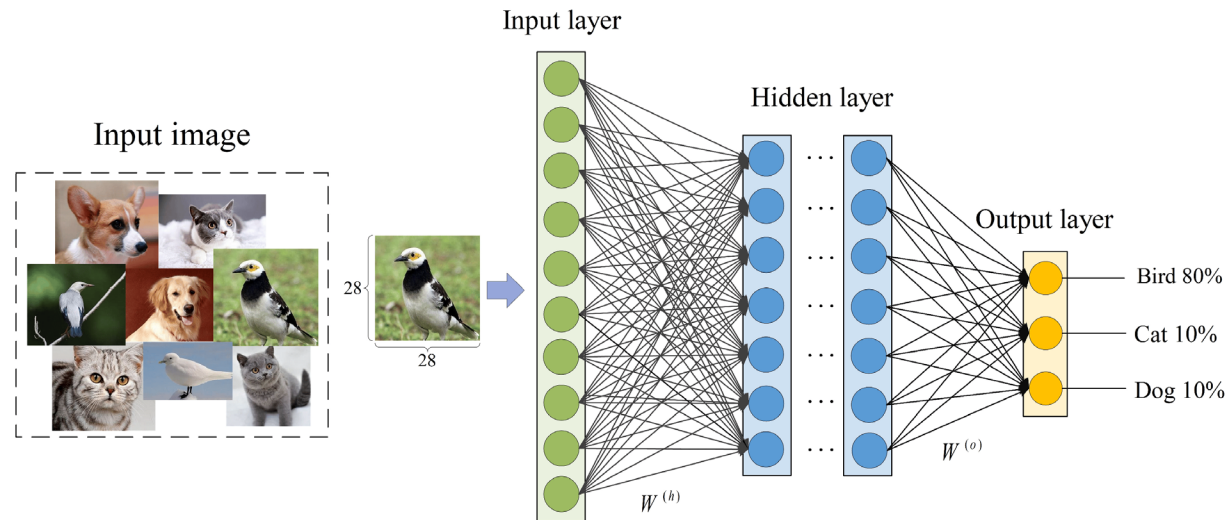


ВИКОРИСТАНІ ТЕХНОЛОГІЇ

- VSCode
- PyCharm
- Git/GitHub
- Google
- Colaboratory
- FastAPI
- Telebot
- PyTest
- HTML
- CSS
- Docker
- Matplotlib
- J-notebook
- Numpy
- Pandas
- Scikit-learn
- Seaborn
- Tensorflow
- Keras

АРХІТЕКТУРА МОДЕЛІ

Попередньо навчена згоркова нейронна мережа VGG16 для класифікації зображень (CIFAR-10)



Переваги VGG16:

Простота структури: VGG16 має відносно просту та регулярну архітектуру, де кожен згортковий блок складається з послідовності однакових згорткових шарів з невеликими 3x3 фільтрами та пулінговими шарами. Це сприяє легшій реалізації та зрозумінню моделі.

Глибока архітектура: VGG16 має 16 згорткових та повнозв'язних шарів, що робить її досить глибокою моделлю. Глибокі мережі здатні до вивчення більш складних та абстрактних ознак, що може покращити їх здатність до класифікації.

Добре передбачувана поведінка: Більш глибокі та регулярні архітектури, такі як VGG16, часто мають добре передбачувану поведінку, що полегшує налаштування та оптимізацію параметрів.

Недоліки VGG16:

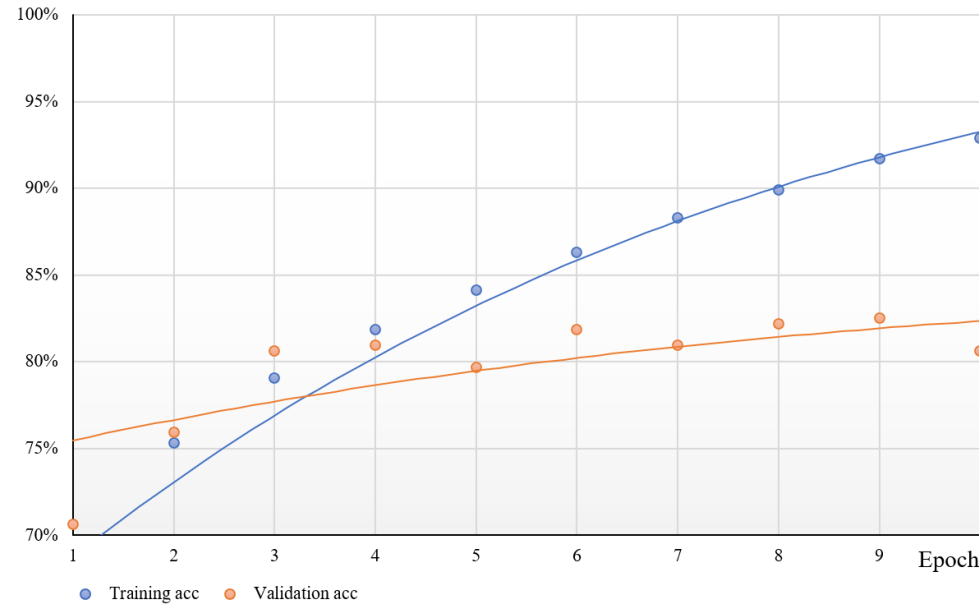
Велика кількість параметрів: Через глибоку архітектуру VGG16 має значну кількість параметрів, що може вимагати велику обчислювальну потужність та ресурси для навчання.

Велика обчислювальна складність: Згорткові пари з меншою розмірністю фільтрів призводять до більшої кількості операцій згортки, що може збільшити час навчання та інференсу.

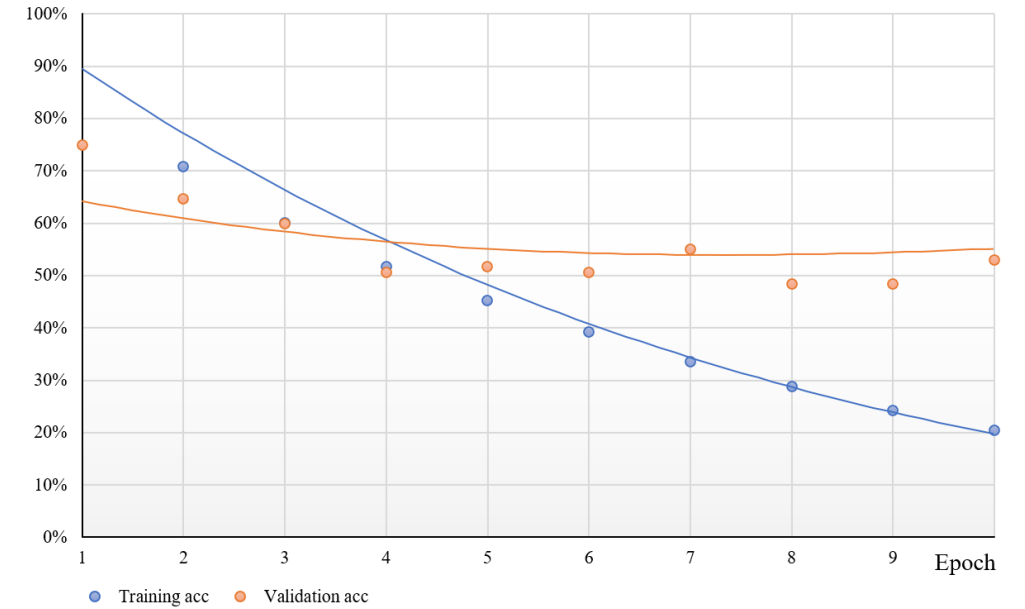
Overfitting: Завдяки своїй глибині та кількості параметрів, VGG16 може бути схильною до перенавчання на невеликих датасетах, якщо не використовувати відповідні методи регуляризації та зменшення розмірності.

ТОЧНІСТЬ ТА РЕЗУЛЬТАТИ

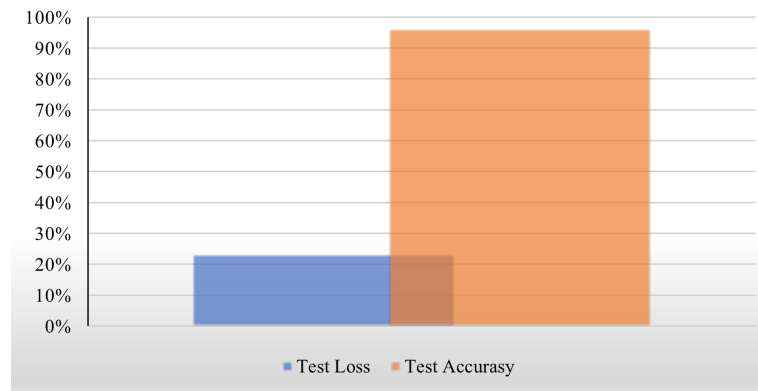
Training and validation accuracy



Training and validation loss



Test Results



РЕАЛІЗАЦІЯ BACKEND ІЗ FAST API

FastAPI - це сучасний веб-фреймворк для створення високопродуктивних API

Переваги FastAPI:

Швидкість: FastAPI використовує техніки оптимізації та асинхронного програмування, що дозволяє забезпечити високу продуктивність і низьку латентність відповідей.

Автоматична документація: FastAPI генерує документацію API автоматично на основі анотацій типів даних та коментарів. Це робить процес розробки і документування API більш простим і ефективним.

Система типізації: використовує анотації типів даних, які допомагають визначити вхідні та вихідні дані, а також забезпечують перевірку їх правильності під час виконання.

Асинхронність: FastAPI підтримує асинхронне програмування, що дозволяє обробляти багато запитів одночасно та ефективно використовувати ресурси сервера.

Інтуїтивний синтаксис: Синтаксис FastAPI дуже зручний та інтуїтивний, особливо для розробників, які вже знайомі з мовою Python.

Недоліки FastAPI:

Вищий поріг входження: FastAPI може бути важким для початківців.

Обмежена екосистема: Екосистема FastAPI може бути менш розвиненою порівняно з іншими фреймворками, такими як Flask або Django.

Молодий фреймворк: Оскільки FastAPI є молодим фреймворком, може виникнути деяка невизначеність або зміни у майбутніх версіях, що може вплинути на стабільність та сумісність.

Менша кількість ресурсів та документації: Порівняно з більш популярними фреймворками, FastAPI може мати меншу кількість онлайн-документації, статей та пакетів спільноти.

ВЗАЄМОДІЯ З КОРИСТУВАЧЕМ



WEB-ВЕРСІЯ
ЗАСТОСУНКУ



FAST API
SWAGGER



TELEGRAM
BOT

ПОКРИТТЯ ПРОЕКТУ ТЕСТАМИ

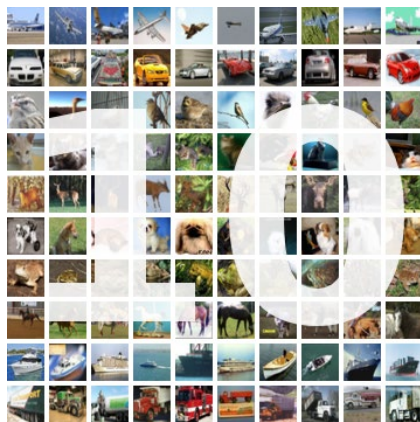
Coverage report: 98%

coverage.py v7.2.7, created at 2023-08-01 13:40 +0300

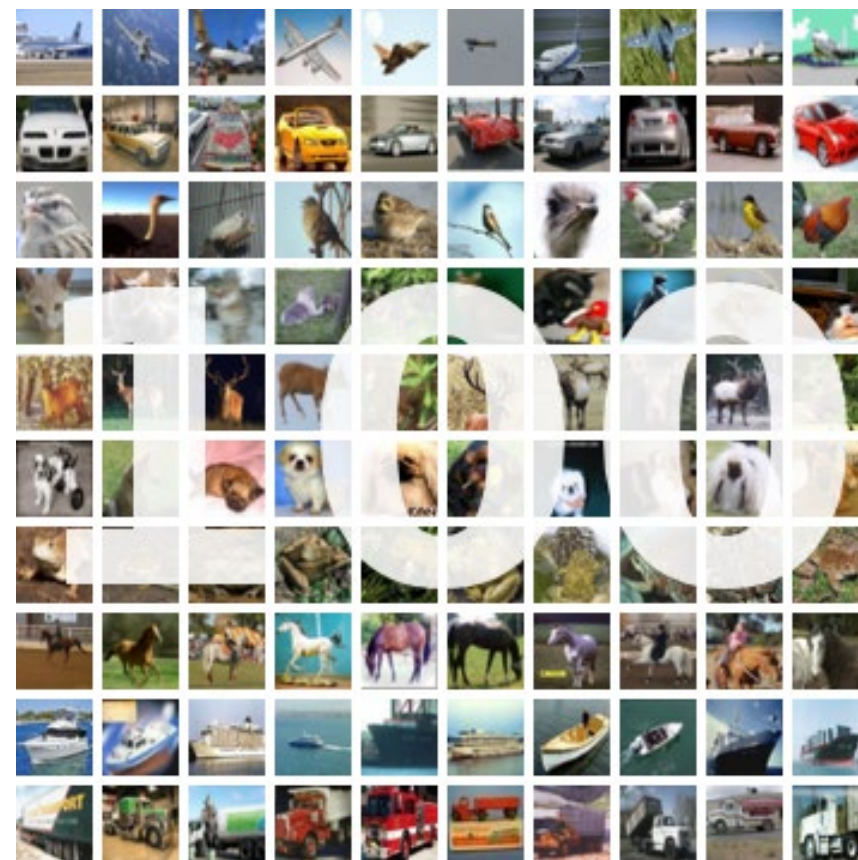
Module	statements	missing	excluded	coverage
main.py	21	2	0	90%
model__init__.py	0	0	0	100%
model\components.py	22	0	0	100%
tests__init__.py	0	0	0	100%
tests\test_components.py	22	0	0	100%
tests\test_main.py	17	0	0	100%
Total	82	2	0	98%

coverage.py v7.2.7, created at 2023-08-01 13:40 +0300

ПЕРСПЕКТИВИ ТА РОЗВИТОК



CIFAR-10



CIFAR-100

ВИСНОВКИ

За час виконання проекту:

- Проведено аналіз предметної галузі та існуючих веб-сервісів на тему проекту.
- Створено згорткову нейронну мережу для класифікації зображень з використанням датасету CIFAR-10.
- Навчено модель для розпізнавання 10 класів зображень (літаки, автомобілі, птахи, коти, олені, собаки, жаби, коні, кораблі та вантажівки)
- Реалізовано взаємодію із користувачем для її використання.
- Розглянуті перспективи її подальшого розвитку.
- Створено веб-сервіс, що дає користувачеві змогу класифікувати зображення за допомогою згорткових нейронних мереж.

Abstract geometric lines in a light beige color, forming various polygons and intersecting patterns on the left side of the slide.

ДЯКУЮ ЗА УВАГУ