

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних систем та мереж

Пояснювальна записка
до кваліфікаційної роботи бакалавра
за спеціальністю 123 «Комп'ютерна інженерія»

на тему: Методи розпізнавання простих
об'єктів, представленими двовимірними монохромними
зображеннями

Проектував	_____	В. Б. Грязнов
Керівник роботи	_____	А. О. Сенько
Нормоконтроль	_____	Д. І. Кузнецов
Завідувач кафедри	_____	А. І. Купін

Криворізький національний університет
 Факультет інформаційних технологій
 Кафедра комп'ютерних систем та мереж

Ступінь вищої освіти
 Спеціальність

бакалавр
 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри, голова циклової комісії

_____ А. І. Купін

“ ___ ” _____ 20__ року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Грязнов Вадим Борисович

(прізвище, ім'я, по батькові)

1. Тема роботи _____ Методи розпізнавання простих об'єктів, представлених
двовимірними монохромними зображеннями

керівник роботи _____ Сенько А. О. _____,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ ___ ” _____ 20__ року №__

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи _____

Набір даних для тренування, валідації та тестування нейронної
мережі

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
 розробити) _____

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових
 креслень) _____ Презентація PowerPoint

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1			
2			
3			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1	Пошук літератури та попередніх досліджень	11.03.24 - 19.03.24	Виконав
2	Сортування знайденої інформації	22.03.24	Виконав
3	Тестування та вибір методів дослідження	23.03.24 – 28.03.24	Виконав
4	Написання першого розділу	02.04.24- 09.04.24	Виконав
5	Вибір існуючих інструментів	17.04.24	Виконав
6	Написання другого розділу	28.04.24	Виконав
7	Створення моделі згорткової мережі, процес навчання	03.05.24	Виконав
8	Аналіз отриманих результатів	12.05.24	Виконав
9	Написання третього розділу	19.05.24	Виконав
10	Написання пояснювальної записки та загальне оформлення дипломної роботи	27.05.24	Виконав

Студент _____
(підпис) (прізвище та ініціали)Керівник роботи _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 60 сторінки, 21 рисунок, 4 таблиць, 15 використаних джерел.

Об'єкт дослідження: Монохромні зображення простих об'єктів.

Мета роботи: Вдосконалення обчислювальних методів розпізнавання простих об'єктів на двовимірних монохромних зображеннях.

Перший розділ присвячений теоретичному аналізу існуючих методів обробки зображень, включаючи класичні підходи та сучасні методи машинного навчання. У другому розділі розглянуті потужності згорткової CNN моделі, її недоліки та переваги. Третій розділ присвячений розробці та навчанню нейронної мережі для считування тексту та аналізу отриманих результатів.

					КНУ.РБ.123.24.06.Р			
Змн.	Арк.	№ документа	Підпис	Дата	Реферат	Літера	Аркуш	Аркушів
Розробив	Грязнов							
Перевірив								
Н.контроль	Кузнецов					КІ-20		
Затвердив	Купін							

ЗМІСТ

РЕФЕРАТ	6
ЗМІСТ	7
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП	10
РОЗДІЛ 1 МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ	12
1.1 Огляд сучасних методів обробки зображень	12
1.1.1 Класичні методи обробки зображень	12
1.1.2 Методи машинного навчання в обробці зображень.....	14
1.2 Глибоке навчання у задачах комп'ютерного зору.....	16
1.2.1 Згорткові нейронні мережі	16
1.2.2 Рекурентні нейронні мережі (RNN) і їх застосування у комп'ютерному зорі	19
1.2.3 Порівняння та актуальність CNN	20
1.3 Проблеми розпізнавання монохромних зображень	21
1.3.1 Низька роздільна здатність.....	21
1.3.2 Відсутність колірної інформації.....	22
РОЗДІЛ 2 ЗГОРТКОВА НЕЙРОННА МЕРЕЖА CNN	25
2.1 . Характеристика, типи та можливості моделі CNN	25
2.2 . Процес згортки	26
2.3 . Пулінгові шари	30
2.4 . Функції втрат	33
2.5 . Активаційні функції	34
Висновок до другого розділу	35

					КНУ.РБ.123.24.06.3			
Змн.	Арк.	№ документа	Підпис	Дата				
Розробив		Грязнов			ЗМІСТ	Літера	Аркуш	Аркушів
Перевірив								
Н.контроль		Кузнецов			КІ-20			
Затвердив		Купін						

РОЗДІЛ 3 ПОБУДОВА МОДЕЛІ ДЛЯ ЧИТАННЯ ТЕКСТУ	36
3.1. Підготовка зображення	36
3.2. MNIST	38
3.3. Навчання моделі	40
3.4 Оцінка моделі	41
ВИСНОВОК.....	45
ПЕРЕЛІК ПОСИЛАНЬ	47
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ	49

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**ПК** – персональний комп'ютер.**CNN** – згорткова нейронна мережа.**MNIST**- Modified National Institute of Standards and Technology**ОС** – операційна система.**AR/VR** - Розширена та віртуальна реальність.**ПЗ** – програмне забезпечення.**ШІ** – штучний інтелект.**OCR** – оптичне розпізнавання символів

					КНУ.РБ.123.24.06.ПУП			
Змн.	Арк.	№ документа	Підпис	Дата	ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	Літера	Аркуш	Аркушів
Розробив	Грязнов							
Перевірив								
Н.контроль	Кузнецов					КІ-20		
Затвердив	Купін							

ВСТУП

В епоху цифровізації та широкого впровадження автоматизації, здатність ефективно аналізувати великі обсяги візуальної інформації є критично важливою для багатьох сфер людської діяльності. Центральним аспектом такого аналізу є розпізнавання об'єктів, що дозволяє комп'ютерним системам ідентифікувати та класифікувати різні об'єкти на зображеннях з мінімальним людським втручанням. Значущість цього процесу посилюється у зв'язку з розвитком технологій машинного зору та їх застосуванням у таких областях, як автоматизоване виробництво, безпілотний транспорт, системи безпеки та медицина.

Ця дипломна робота зосереджується на дослідженні та вдосконаленні обчислювальних методів розпізнавання простих об'єктів на двовимірних монохромних зображеннях. Монохромні зображення, попри свою здавалося б простоту, надають унікальні виклики через обмежену кількість інформації, яка має бути оброблена для успішного розпізнавання. Обрані методи базуються на останніх досягненнях у галузі машинного навчання та комп'ютерного зору, включаючи нейронні мережі, глибинне навчання, а також класичній обробці зображень.

Об'єктом дослідження є монохромні зображення простих об'єктів, де ключовими є виклики, пов'язані з обмеженнями у візуальній інформації та необхідністю її точної інтерпретації.

Предметом роботи є застосування методів глибокого навчання, зокрема згорткових нейронних мереж (CNN) та методів класичної обробки зображень, для аналізу та ідентифікації об'єктів.

					КНУ.РБ.123.24.06.В			
Змн.	Арк.	№ документа	Підпис	Дата				
Розробив	Грязнов				ВСТУП	Літера	Аркуш	Аркушів
Перевірив								
Н.контроль	Кузнецов				КІ-20			
Затвердив	Купін							

Головною метою роботи є вдосконалення існуючих методик розпізнавання, зокрема покращення точності та ефективності в реальному часі. Особлива увага приділяється алгоритмам, які можуть ефективно працювати з великими даними та в умовах обмежених ресурсів, що робить їх придатними для використання в мобільних пристроях та на периферійних обчислювальних пристроях.

Для досягнення поставлених цілей, робота включає теоретичне дослідження існуючих методів, їх аналіз та модифікацію, а також практичну реалізацію прототипи обраного методу з використанням програмних засобів. Результати дослідження мають бути валідовані через низку експериментів, які демонструють покращення в точності та швидкості обробки зображень.

Таким чином, ця дипломна робота спрямована на значне покращення можливостей обчислювальних систем у сфері розпізнавання зображень, що буде сприяти технічному прогресу і може знайти застосування в широкому спектрі прикладних областей.

РОЗДІЛ 1 МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ

1.1 Огляд сучасних методів обробки зображень

У загальному випадку алгоритм розпізнавання передбачає перевірку та поступове висування гіпотез. Порядок висування нових гіпотез залежить від результатів перевірки попередніх. Кожна гіпотеза зазвичай має числову оцінку або результат порівняльної операції в OCR системах. Гіпотези висуваються одна за одною, потім збираються в список і сортуються на основі їхніх оцінок. Остаточний вибір гіпотези робиться з урахуванням контексту, а також, можливо, з використанням додаткових знань.



Рисунок 1.1 - Структурна схема розпізнавання зображення

1.1.1 Класичні методи обробки зображень

Класичні методи включають ряд базових технік, які не вимагають складного машинного навчання і часто застосовуються для попередньої обробки або поліпшення якості зображень.

Фільтрація

Фільтрація є одним із найпоширеніших методів обробки зображень, який використовується для видалення шуму, згладжування зображень, підвищення чіткості або виділення певних особливостей зображення. Існують різні типи фільтрів, зокрема просторові та частотні фільтри.

Просторові фільтри працюють безпосередньо з пікселями зображення, і найчастіше використовуються такі фільтри, як середній, медіанний і гауссовий.

Середній фільтр застосовує усереднення значень пікселів у певному вікні, що дозволяє згладити зображення і зменшити шум. Проте, цей метод може розмивати деталі.

КНУ.РБ.123.24.06.МОЗ

Змн.	Арк.	№ документа	Підпис	Дата			
					МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ		
Розробив	Грязнов						
Перевірив							
Н.контроль	Кузнєцов				КІ-20		
Затвердив	Купін						

Медіанний фільтр замінює значення кожного пікселя медіанним значенням у вікні, що дозволяє ефективно видаляти імпульсні шуми (спекл), зберігаючи краще різкі краї об'єктів.

Гауссовий фільтр використовує гауссове розподілення для згладжування зображення, що дозволяє зберегти більше деталей у порівнянні з середнім фільтром.

Частотні фільтри працюють у частотній області, після перетворення зображення методом Фур'є. Це дозволяє фільтрувати певні частоти для видалення періодичних шумів або виділення текстур.

Морфологічні операції

Морфологічні операції працюють з формою об'єктів на зображенні і використовуються для виділення структурних особливостей. Основні операції включають ерозію, дилатацію, відкриття та закриття.

- Ерозія зменшує розміри об'єктів, видаляючи пікселі на краях.
- Дилатація збільшує розміри об'єктів, додаючи пікселі до країв.
- Відкриття видаляє невеликі об'єкти.
- Закриття (комбінація дилатації та ерозії) заповнює дрібні отвори в об'єктах.

Морфологічні операції зазвичай застосовуються до бінарних зображень, але можуть бути розширені і до сіро масштабних зображень.

Таблиця. 1.1 Порівняння методів

Методи	Переваги	Недоліки
Фільтрація	- Простота реалізації	- Може втрачати деталі
	- Висока швидкість обробки	- Обмежена складність для складних задач
Рівні яскравості	- Простота використання та реалізації	- Можливість створення артефактів
	- Контроль за параметрами яскравості та контрасту	- Нелінійний вплив на зображення
Гістограмне вирівнювання	- Значне покращення контрасту зображення	- Можливість перевищення інтенсивності, створення артефактів
	- Простота реалізації	- Можливе збільшення шуму
Морфологічні операції	- Ефективність при аналізі форми та структури об'єктів	- Можуть створювати артефакти на краях
	- Простота векторизації та виділення об'єктів	- Вимоги до чіткості об'єктів

Рівні яскравості

Зміна рівнів яскравості та контрасту зображення є важливим кроком в обробці зображень, який дозволяє підвищити якість зображення та покращити

видимість деталей. Це може бути зроблено шляхом лінійного або нелінійного масштабування значень яскравості пікселів.

Лінійне масштабування включає множення всіх значень пікселів на константу, що дозволяє збільшити або зменшити яскравість зображення. Зміна контрасту може бути досягнута шляхом розширення або стискання діапазону яскравостей.

Нелінійні методи, такі як гамма-корекція, використовують степеневі функції для коригування яскравості, що дозволяє більш гнучко контролювати яскравість і контраст.

Гістограмне вирівнювання

Гістограмне вирівнювання – це метод, що використовується для покращення контрасту зображення шляхом рівномірного розподілу інтенсивностей пікселів. Це досягається шляхом перетворення гістограми оригінального зображення у більш рівномірний вигляд.

Процес включає обчислення кумулятивної гістограми зображення, що потім використовується для перетворення оригінальних значень пікселів. Це дозволяє виділити деталі в областях з низьким контрастом.

1.1.2 Методи машинного навчання в обробці зображень

Машинне навчання в обробці зображень є одним із найактивніших та найперспективніших напрямків сучасної науки і технологій. Застосування методів машинного навчання дозволяє автоматизувати та значно покращити процеси аналізу, розпізнавання та класифікації зображень.



Рисунок 1.2 - Відмінність машинного навчання

Сучасні підходи до обробки зображень часто включають методи машинного навчання, які надають більш гнучкі та потужні засоби для розпізнавання об'єктів:

Випадкові ліси: Енсамбль дерев рішень, який використовується для класифікації зображень на основі різних характеристик екстрагованих з даних. Ці моделі добре справляються з переоснащенням і забезпечують високу точність.

Підтримуючі векторні машини (SVM): Цей метод ефективний при роботі з високорозмірними просторами ознак, які часто зустрічаються у зображеннях. SVM можуть класифікувати зображення, використовуючи гіперплощину для розділення класів з максимальним зазором.

Нейронні мережі: Особливу увагу варто приділити згортковим нейронним мережам (CNN), які є стандартом де-факто для задач комп'ютерного зору, зокрема для розпізнавання образів на зображеннях. CNN автоматично виявляють важливі ознаки без потреби в їх ручному визначенні та можуть ефективно масштабуватися для роботи з великими наборами даних.

Розуміння та використання цих методів дозволяє значно підвищити ефективність систем розпізнавання зображень.

Розпізнавання зображення - це складний процес, що включає в себе виконання різних завдань для отримання кінцевого результату у вигляді розуміння комп'ютером що зображено на рисунку. Деякі комерційні системи використовують набір алгоритмів, інші можуть користуватися клітинними автоматами, або об'єднують кілька методів для отримання більш точних результатів. Наприклад, одна частина системи може базуватися на нейронних мережах для розпізнавання тексту, інша - на акумулюванні статистики та її подальшому використанні для формування результату. Комплексний підхід, що поєднує різні заходи та алгоритми, зазвичай дає кращі результати, ніж використання окремих методів. Клітинні автомати можуть бути важливою частиною такого комплексу, оскільки вони дозволяють виконувати обчислення швидко та ефективно, особливо в області обробки зображень, завдяки простоті правил, на яких вони базуються, та можливості паралельної обробки даних.



Рисунок 1.3 - Символи англійського алфавіту

Алгоритм спирається на ідентифікацію характерних ліній, що утворюють символи, навіть якщо ці лінії деформовані, але їх взаємне розташування залишається незмінним. Цей метод виявляється особливо ефективним при розпізнаванні символів, які зазнали деформації або частково накладені один на одного. Наприклад, зображення демонструє викривлені символи "EP", символи "N" та "O", вкладені один в одного, а також накладені символи "VZ". Людина без зусиль може розпізнати ці символи, що свідчить

про перевагу цього методу в порівнянні з традиційними системами розпізнавання.

1.2 Глибоке навчання у задачах комп'ютерного зору

Глибоке навчання здобуло значну популярність у галузі комп'ютерного зору завдяки своїй здатності ефективно вирішувати складні задачі, які були важко доступні для попередніх поколінь алгоритмів. Особливо великий вплив глибоке навчання мало на розвиток методів автоматичного розпізнавання образів на зображеннях.

1.2.1 Згорткові нейронні мережі

Згорткові нейронні мережі (CNN) можна класифікувати за типом вхідних даних та завдань, які вони вирішують. Вони спеціалізуються на обробці зображень, вивчаючи ознаки та шаблони, такі як краї, форми або текстури, для класифікації зображень. CNN широко застосовуються у задачах комп'ютерного зору, таких як розпізнавання об'єктів або сцен. Важливою перевагою CNN є їх здатність автоматично вивчати ваги фільтрів для виявлення ознак на зображеннях, що робить їх ефективними для обробки зображень.

Таблиця. 1.2 Порівняння CNN

Характеристика	1D CNN	2D CNN	3D CNN
Застосування	Аналіз часових рядів, аудіосигналів	Класифікація зображень, сегментація зображень	Класифікація відео, аналіз відеоданих
Кількість вимірів	1 (одновимірний вхід)	2 (двовимірний вхід, зображення)	3 (тривимірний вхід, відео)
Фільтри	1D фільтри	2D фільтри	3D фільтри
Розмір фільтрів	Ширина \times 1 (довжина фільтра \times 1)	Ширина \times Висота (довжина фільтра \times висота)	Ширина \times Висота \times Глибина (довжина фільтра \times висота \times глибина)
Поширення (strides)	1D поширення	2D поширення	3D поширення
Вибірковість (pooling)	1D pooling	2D pooling	3D pooling
Представлення даних	Вектори, часові ряди	Матриці (зображення)	Тривимірні дані (відео)
Застосування в реальному часі	Високо (швидка обробка аудіо)	Залежить від розміру зображення і моделі	Менш ефективно через обробку тривимірних даних

CNN можна використовувати не лише для двовимірних зображень, але й для одновимірних та тривимірних даних. Наприклад, одновимірні дані можуть бути звуковими сигналами, які CNN використовують для розпізнавання тексту, як у моделі Jasper від Nvidia. Тривимірні дані можуть бути відео, де час є третім виміром.

Основні типи CNN:

- 1D CNN: Використовуються для обробки сигналів, таких як часові ряди або аудіо, для виявлення важливих подій або змін у сигналі.
- 2D CNN: Стандартний вибір для класифікації зображень, розпізнавання об'єктів та сегментації зображень. Вони виявляють ключові особливості на зображенні та використовують їх для визначення класу зображення.
- 3D CNN: Застосовуються для класифікації відео, аналізуючи відеодані з урахуванням просторової та часової інформації.

1D CNN - це тип нейронної мережі, де згортка застосовується лише до одного виміру даних. Це означає, що мережа аналізує вхідні дані як послідовність значень, що робить її ідеальною для роботи з часовими рядами або одновимірними сигналами, наприклад, з аудіо даними.

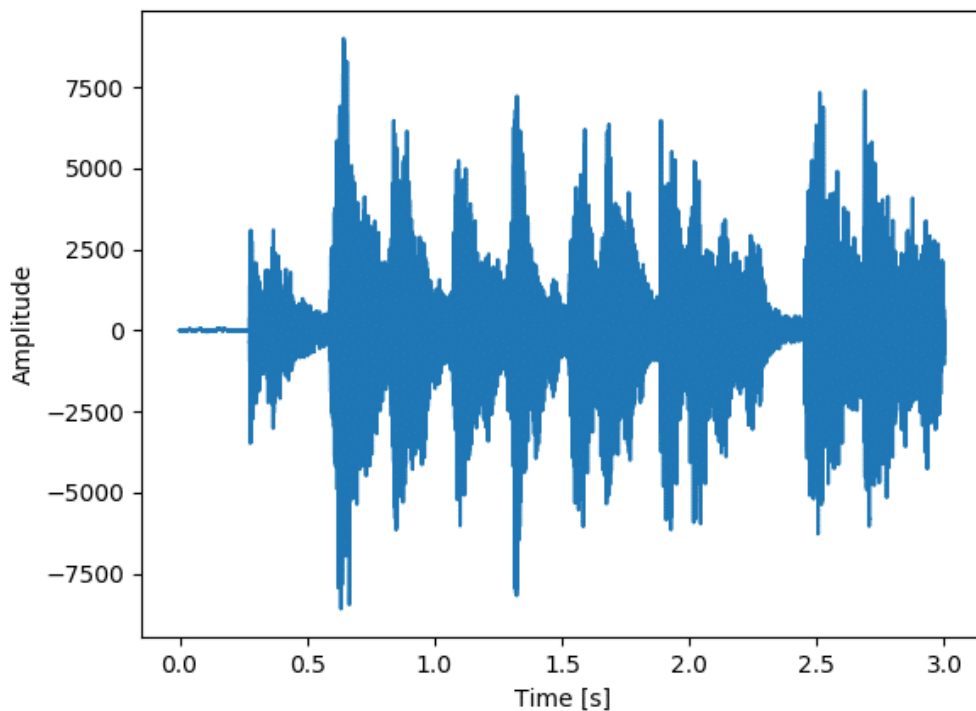


Рисунок 1.4 - Послідовність частотних хвиль

Одновимірні згорткові нейронні мережі (1D CNN) використовуються для обробки аудіофайлів через їх здатність аналізувати послідовні дані. Аудіофайли можна розглядати як часові ряди, де кожен відбиток часу містить інформацію про амплітуду звуку на певний момент часу. Такі дані ідеально підходять для застосування одновимірних згорткових мереж.

Одні з основних способів використання 1D CNN в аудіоаналітиці включають класифікацію звуків, розпізнавання мовлення, виявлення звуків або патернів у звуці. Наприклад, для класифікації жанрів музики модель може

використовувати 1D CNN для виявлення характерних акустичних шаблонів у музичних фрагментах.

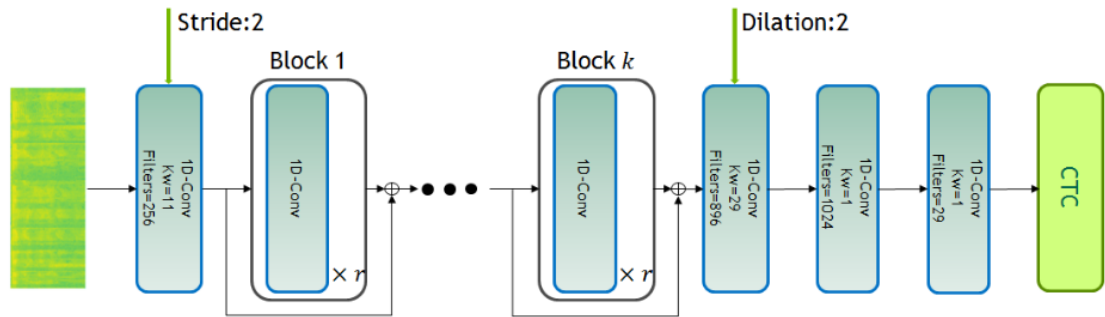


Рисунок 1.5 - Перетворення аудіо файлів в текст за допомогою Jasper

Одновимірні згорткові нейронні мережі (1D CNN) широко застосовуються в аудіоаналізі завдяки їх здатності ефективно обробляти та аналізувати часові ряди, якими є аудіо сигнали. Вони можуть автоматично витягувати характерні ознаки з аудіо даних, такі як тональні, ритмічні та тембральні властивості, що є важливими для завдань класифікації, розпізнавання та виявлення подій в аудіо. Однією з типових задач для одновимірних згорткових мереж є аналіз часових рядів, таких як прогнозування погоди або виявлення аномалій у фізіологічних сигналах. Вони також застосовуються у вирішенні задач обробки мовлення, наприклад, класифікації текстів або розпізнаванні мовлення.

Одновимірні згорткові мережі можуть бути дуже ефективними у вирішенні задач обробки послідовних даних завдяки їхній здатності виявляти важливі шаблони у вхідних даних.

Тривимірні згорткові нейронні мережі (3D CNN) — це варіант згорткових нейронних мереж, які використовуються для обробки тривимірних даних, таких як відео. Вони розширюють концепцію згорткових шарів на третій вимір, дозволяючи враховувати просторову інформацію разом із часою.

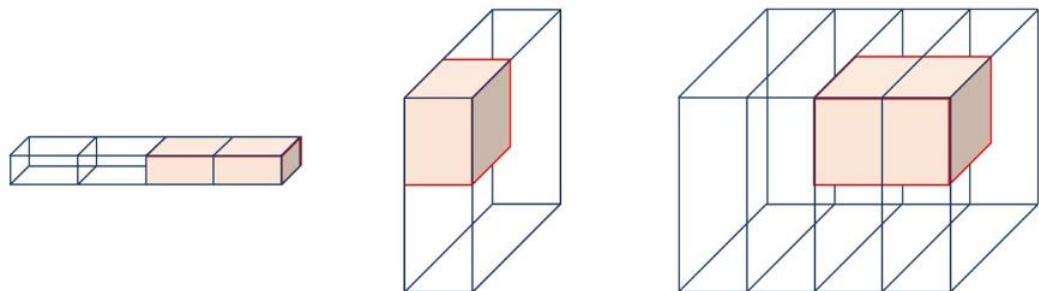


Рисунок 1.6 - Приклади застосування тривимірних нейронних мереж

Основні концепції 3D CNN подібні до 2D CNN, але з деякими відмінностями. Замість зображень у 2D CNN, у 3D CNN використовуються відеофрагменти або тривимірні об'єкти. Згорткові шари у 3D CNN працюють з тривимірними фільтрами, які скользять по трьом вимірам: ширині, висоті та глибині об'єкта.

3D CNN добре підходять для завдань, де важлива інформація міститься у трьох вимірах, наприклад, в аналізі відео. Вони можуть використовуватися

для класифікації відео, виявлення об'єктів у відеопотоці, аналізу руху та інших задач, де важливо враховувати інформацію у тривимірному просторі та часі.

Тривимірні згорткові нейронні мережі (3D CNN) — це тип нейронних мереж, що використовується для обробки тривимірних даних, наприклад, відео. Вони розширюють концепцію згорткових шарів на третій вимір, що дозволяє їм аналізувати просторову інформацію разом із часою.

Основна ідея 3D CNN подібна до 2D CNN, але згорткові шари працюють з тривимірними фільтрами, які скользять по трьом вимірам: ширині, висоті та глибині об'єкта. Це дозволяє їм ефективно виявляти просторові особливості об'єктів у тривимірному просторі, такі як форма, текстура та розмір, а також часові залежності, такі як рух.

Застосування 3D CNN включає класифікацію відео, розпізнавання дій у відео, аналіз медичних зображень (наприклад, рентгенівських знімків) та інші задачі, де важлива тривимірна інформація. Вони дозволяють ефективно моделювати просторово-часові залежності в даних і використовувати цю інформацію для точного розпізнавання та класифікації.

1.2.2 Рекурентні нейронні мережі (RNN) і їх застосування у комп'ютерному зорі

Основи RNN: Рекурентні нейронні мережі (RNN) оптимізовані для обробки послідовних даних, як-от часові ряди або текст. В комп'ютерному зорі RNN можуть бути застосовані для задач, де зображення або відео розглядаються як послідовність, наприклад, при аналізі відео в реальному часі.

Застосування RNN: Одним із прикладів застосування RNN у комп'ютерному зорі є розпізнавання дій на відео, де послідовність кадрів аналізується для ідентифікації певних дій.

Кінцеві системи: Завдяки глибокому навчанню, сучасні системи комп'ютерного зору здатні автоматизувати багато складних процесів, які раніше вимагали значних людських зусиль. Ці системи можуть включати розпізнавання об'єктів, класифікацію зображень, детекцію подій та багато іншого.

Переваги і виклики: Хоча глибоке навчання і принесло значні поліпшення у точності та ефективності систем комп'ютерного зору, воно також ставить нові виклики, такі як потреба в великих наборах даних для тренування і високі вимоги до обчислювальних ресурсів.

Ці технології відкривають нові можливості для покращення та розширення функціональності обчислювальних систем, сприяючи розвитку інноваційних рішень у багатьох галузях, включаючи автоматизацію, моніторинг безпеки, розумні транспортні системи, та медичну візуалізацію.

1.2.3 Порівняння та актуальність CNN

Згорткові нейронні мережі (CNN) та нейронні мережі прямого поширення (FNN) відрізняються основним чином у використанні та обробці даних. FNN використовуються для виявлення шаблонів та взаємозв'язків у даних, що подаються в одновимірному вигляді, тоді як CNN оптимізовані для роботи з двовимірними даними, такими як зображення.

FNN відмінно справляються з аналізом послідовностей даних, таких як текст або звуки, оскільки вони можуть "пам'ятати" попередні вхідні дані та використовувати цю інформацію для подальшого прийняття рішень. Однак вони можуть виявити складність у вирішенні завдань, що пов'язані з обробкою зображень, оскільки для цього потрібно враховувати просторову структуру даних.

Навпаки, CNN більш ефективні для обробки зображень, оскільки вони можуть врахувати просторові залежності між пікселями. За допомогою спеціальних шарів, таких як згорткові та пулінгові шари, вони можуть впізнавати шаблони, такі як краї та текстури, що робить їх ефективними для задач обробки зображень.

Отже, вибір між CNN та FNN залежить від типу даних, з якими ви працюєте, та ваших конкретних завдань. Якщо ви маєте справу з послідовністю даних, то FNN може бути кращим варіантом, а для обробки зображень варто розглянути використання CNN.

Згорткові нейронні мережі (CNN) є потужним інструментом в області комп'ютерного зору та обробки зображень, але їх застосування виходить далеко за межі цього. Ось кілька основних напрямків, де CNN знаходять своє застосування:

1. Обробка зображень

- Класифікація зображень: CNN широко використовуються для розпізнавання та класифікації зображень. Наприклад, вони можуть визначати, чи зображення містить kota чи собаку.
- Розпізнавання об'єктів: CNN можуть визначати і локалізувати об'єкти на зображеннях, такі як автомобілі, пішоходи, дорожні знаки тощо.
- Сегментація зображень: Згорткові нейронні мережі (CNN) використовуються для класифікації кожного пікселя в зображенні, що дозволяє виділяти окремі об'єкти або області на зображенні.

2. Обробка відео

- Класифікація відео: 3D CNN аналізують відеодані, враховуючи просторову та часову інформацію, для класифікації відео за різними категоріями.
 - Аналіз дій: CNN можуть визначати дії, які виконуються у відео, такі як біг, стрибки, ходьба тощо.

3. Обробка аудіо

- Розпізнавання мови: 1D CNN використовуються для аналізу аудіозаписів мовлення, виявлення конкретних звуків або слів для подальшого розпізнавання мови.

- Класифікація музики: 1D CNN аналізують музичні треки, визначаючи характеристики, що є типовими для певних музичних жанрів.

4. Обробка тексту

- Аналіз тональності: CNN можуть використовуватися для аналізу тексту, визначення його тональності (позитивний, негативний або нейтральний).

- Класифікація тексту: CNN можуть класифікувати текстові дані за різними категоріями, такими як новини, спорт, розваги тощо.

5. Біомедичні зображення

- Аналіз медичних зображень: CNN допомагають лікарям аналізувати медичні зображення, такі як рентгенівські знімки, МРТ, КТ, для виявлення захворювань або патологій.

- Сегментація тканин: CNN використовуються для автоматичного виділення і сегментації різних тканин або органів на медичних зображеннях.

6. Автономні транспортні засоби

- Системи комп'ютерного зору: CNN використовуються в автономних автомобілях для розпізнавання об'єктів, дорожніх знаків, пішоходів, транспортних засобів тощо, що забезпечує безпечну навігацію та ухвалення рішень на дорозі.

7. Розширена та віртуальна реальність

- Відстеження рухів: CNN застосовуються для відстеження рухів голови, рук та інших частин тіла користувача, що дозволяє створювати інтерактивні та реалістичні AR/VR-досвіди.

Згорткові нейронні мережі демонструють велику ефективність у різних галузях, забезпечуючи точність і надійність в задачах розпізнавання та класифікації завдяки своїй здатності виявляти та використовувати ключові особливості вхідних даних.

1.3 Проблеми розпізнавання монохромних зображень

1.3.1 Низька роздільна здатність

Низька роздільна здатність є однією з основних проблем, що виникають при роботі з монохромними зображеннями. Це означає, що зображення мають обмежену кількість пікселів або діапазон тонів, що впливає на чіткість та деталізацію зображення. Давайте розглянемо цей аспект детальніше:

Проблема:

Втрата дрібних деталей: Монохромні зображення можуть не зберегти дрібні деталі, які можуть бути важливими для розпізнавання об'єктів.

					КНУ.РБ.123.24.06.МОЗ	Арк.
Арк.	№ документа	Підпис	Дата			

Наприклад, на зображенні з низькою роздільною здатністю можуть бути важко розрізнити маленькі текстові символи або інші дрібні деталі.

Розмитість зображення: Обмежена кількість пікселів може призвести до розмитості зображення, особливо у випадку масштабування або збільшення зображення. Це може зробити навіть великі об'єкти на зображенні менш чіткими та менш розпізнаваними.

Рішення:

Використання вищої роздільної здатності: Одним з можливих рішень є отримання зображень з вищою роздільною здатністю, яка може допомогти зберегти більше деталей та покращити якість зображення.

Використання методів розмиття: Деякі методи обробки зображень, такі як фільтри розмиття, можуть допомогти зменшити рівень шуму та покращити вигляд розмитих деталей на зображенні.

Аугментація даних: Додавання штучного шуму або інших форм аугментації даних може допомогти навчити моделі розпізнавати об'єкти на зображеннях з низькою роздільною здатністю.

Низька роздільна здатність може стати серйозним обмеженням у процесі розпізнавання об'єктів на монохромних зображеннях, і вимагає ретельного аналізу та використання відповідних стратегій для її подолання.

1.3.2 Відсутність колірної інформації

Відсутність колірної інформації у монохромних зображеннях може стати викликом при розпізнаванні об'єктів та аналізі зображень. Давайте розглянемо цей підпункт детальніше:

Проблема:

Втрата важливої інформації: Кольорова інформація може бути важливою для визначення характеристик об'єктів або їх контексту. Наприклад, колір може допомагати в розпізнаванні різних видів фруктів, транспортних засобів або знаків.

Ускладнення розпізнавання: Відсутність колірної інформації може ускладнити процес розпізнавання об'єктів, особливо якщо об'єкти мають схожі форми або текстури, але відрізняються за кольором.

Рішення:

Використання інших ознак: Поміж іншими характеристиками, такими як текстура, форма, розмір та контекст, можна використовувати для розпізнавання об'єктів на монохромних зображеннях.

Використання текстурних ознак: Текстурні характеристики, такі як густина, гладкість або грубість, можуть бути важливими для розпізнавання об'єктів на зображенні та можуть допомогти компенсувати відсутність колірної інформації.

Комбінація з іншими джерелами інформації: Використання інших джерел інформації, таких як текстові описи, метадані або контекстуальні дані, може допомогти компенсувати відсутність колірної інформації та полегшити процес розпізнавання об'єктів.

Врахування відсутності колірної інформації та розробка відповідних стратегій для її компенсації є ключовими аспектами успішного розпізнавання об'єктів на монохромних зображеннях.

1.3.3 Обмеженість контексту

Обмеженість контексту на монохромних зображеннях може ускладнити процес розпізнавання об'єктів та аналізу зображень. Давайте розглянемо цей аспект детальніше:

Проблема:

Втрата просторового контексту: На монохромних зображеннях може бути важко визначити просторовий контекст об'єктів. Це може ускладнити визначення їхнього розміру, розташування та взаємозв'язку з іншими об'єктами на зображенні.

Обмежена інформація про оточуючі об'єкти: Монохромні зображення можуть не передавати детальної інформації про оточуючі об'єкти або фон. Це може ускладнити відрізнення об'єктів від фону або інших об'єктів, що знаходяться поруч.

Рішення:

Використання контекстуальних ознак: Контекстуальні ознаки, такі як розмір, форма або розташування об'єктів відносно один одного, можуть бути використані для покращення розпізнавання об'єктів та аналізу зображень.

Використання інформації про структуру: Врахування структури та організації об'єктів на зображенні може допомогти визначити їхній контекст та взаємозв'язок між ними.

Використання багатомасштабних підходів: Використання методів багатомасштабного аналізу дозволяє аналізувати зображення на різних рівнях деталізації та виявляти зв'язки між об'єктами на різних масштабах.

Подолання обмеженостей контексту на монохромних зображеннях вимагає використання різноманітних підходів та стратегій, спрямованих на отримання більш повної та точної інформації про об'єкти та їхнє оточення.

Висновок до першого розділу

У першому розділі були досліджені основні аспекти розпізнавання об'єктів на монохромних зображеннях, зокрема проблеми низької роздільної здатності, відсутності колірної інформації та обмеженості контексту. Для вирішення цих проблем було надано перевагу за згортковим нейронними мережами (CNN).

Монохромні зображення, хоч і є обмеженими у відображенні деталей та контексту, все ж можуть бути ефективно аналізовані за допомогою глибокого навчання. CNN виявляються особливо перспективними для цього завдання, оскільки вони здатні автоматично виявляти та використовувати локальні шаблони та ознаки на зображеннях.

Використання CNN відкриває можливості для подальшого покращення систем розпізнавання об'єктів на монохромних зображеннях. Зокрема, це може сприяти підвищенню точності розпізнавання, зниженню кількості помилок та підвищенню ефективності системи в цілому.

Отже, на основі аналізу проблем та можливостей, використання CNN виявляється перспективним напрямом для подальшого дослідження та розвитку систем розпізнавання об'єктів на монохромних зображеннях.

					КНУ.РБ.123.24.06.МОЗ	Арк.
	Арк.	№ документа	Підпис	Дата		

РОЗДІЛ 2 ЗГОРТКОВА НЕЙРОННА МЕРЕЖА CNN

2.1 . Характеристика, типи та можливості моделі CNN

CNN складається з декількох типів шарів, кожен з яких виконує специфічну функцію:

Згорткові шари: Ядро цих мереж, згорткові шари використовують набори навчених фільтрів, які "проходять" через вхідне зображення для створення карт ознак. Кожен фільтр автоматично виявляє специфічні ознаки, такі як краї, кути або більш складні текстурні шаблони на різних рівнях складності. Операція згортки зводиться до математичного множення фрагментів зображення на матрицю ваг фільтра, що дозволяє виділити необхідні ознаки.

Активаційні функції: Після кожного згорткового шару застосовують активаційну функцію, таку як ReLU (Rectified Linear Unit), яка додає нелінійність до обробки даних. Це дозволяє CNN ефективно навчатися та моделювати складніші взаємозв'язки у даних.

Пулінгові шари: Шари пулінгу (зазвичай max-pooling) використовуються для зменшення розмірів карт ознак шляхом вибору максимального значення з кожного блоку пікселів. Це допомагає зменшити кількість параметрів і обчислювальне навантаження, збільшуючи при цьому стійкість мережі до невеликих зміщень у вхідних даних.

Повнозв'язні шари: На заключних етапах CNN зазвичай розміщують один або декілька повнозв'язних шарів, які обробляють весь набір ознак, отриманих з попередніх шарів, для виконання класифікації або інших завдань. Тут вся інформація з карт ознак комбінується для вирішення конкретної задачі, такої як розпізнавання об'єктів.

					КНУ.РБ.123.24.06.ЗНМС		
Змн.	Арк.	№ документа	Підпис	Дата			
Розробив	Грязнов				Літера	Аркуш	Аркушів
Перевірив							
Н.контроль	Кузнецов				ЗГОРТКОВА НЕЙРОННА МЕРЕЖА CNN		
Затвердив	Купін						

Навчання CNN

Процедура навчання: CNN навчають за допомогою алгоритму зворотного розповсюдження помилки, який ефективно коригує ваги мережі з метою мінімізації розбіжностей між фактичними результатами та передбаченнями мережі. Для навчання використовуються методи оптимізації, такі як SGD (стохастичний градієнтний спуск), Adam або RMSprop, які допомагають знаходити оптимальні ваги для кожного фільтра в мережі.

Застосування CNN

CNN застосовуються у широкому спектрі задач комп'ютерного зору, включаючи:

Розпізнавання об'єктів: Автоматичне ідентифікація об'єктів на зображеннях, таких як люди, автомобілі або дорожні знаки.

Семантична сегментація: Класифікація кожного пікселя зображення з метою визначення до якого об'єкта або категорії він належить.

Аналіз відео: Використання CNN для аналізу відеопотоків з метою виявлення та класифікації різних подій або активностей.

Згорткові нейронні мережі змінили парадигму в області комп'ютерного зору, пропонуючи високу точність і надійність у різноманітних застосуваннях, від автоматичного водіння до систем спостереження та медичних діагностичних систем.

Переваги CNN: Найбільшою перевагою CNN є їх здатність до самостійного виявлення необхідних ознак без необхідності ручного проектування фільтрів, що було обмеженням традиційних методів обробки зображень.

Структурна схема згорткової нейронної мережі наведена на рис. 1.2.

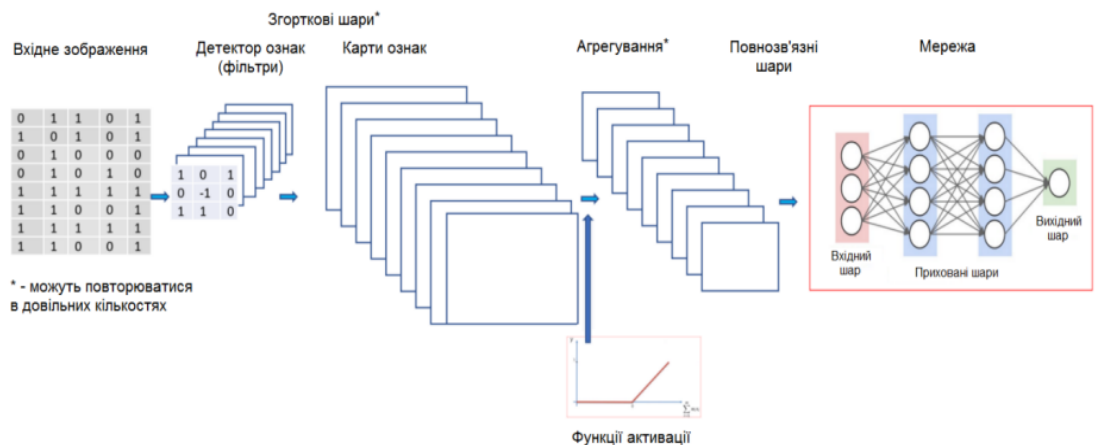


Рисунок 2.1 - Структурна схема побудови згорткової нейронної мережі

2.2 . Процес згортки

Згортка — це процес застосування ядра (фільтра) до кожної ділянки вхідного зображення.

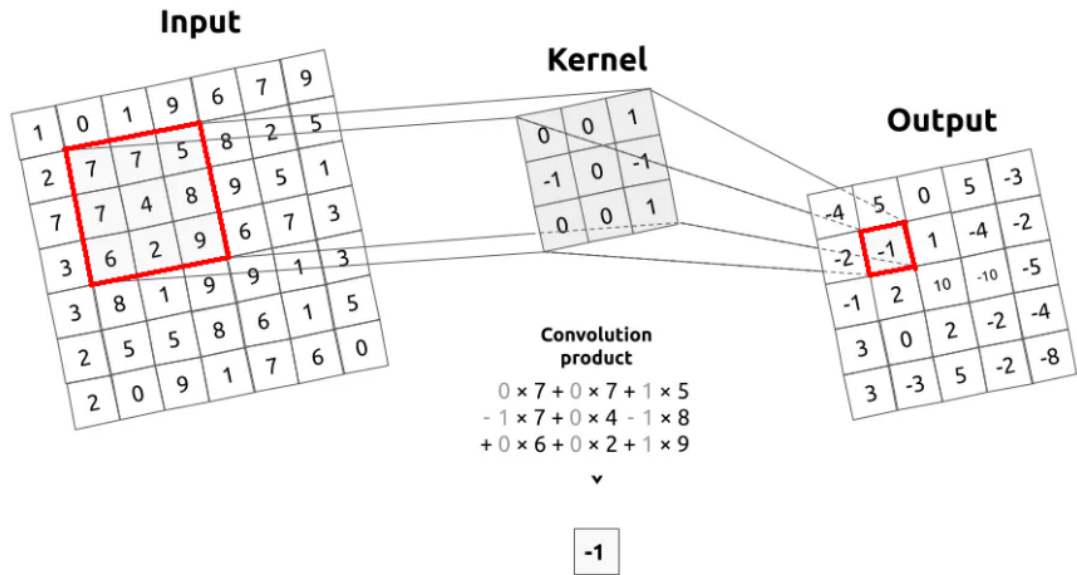


Рисунок 2.2 - Приклад операції згортки

Зображення демонструє основний принцип роботи згорткових нейронних мереж (CNN) — операцію згортки. Це один із ключових етапів обробки зображень у CNN, який використовується для виявлення різноманітних ознак (features) на зображеннях, таких як краї, текстури та інші патерни. Вхідне зображення представлено як матриця розміром 6x6, де кожен елемент матриці відповідає значенню пікселя. Ядро (або фільтр) — це матриця меншого розміру, в цьому випадку 3x3, яка використовується для обробки вхідного зображення. Ядро має задані значення, які будуть застосовані до пікселів вхідного зображення. Вихідна матриця має розмір 4x4, де кожен елемент обчислюється як результат операції згортки відповідної області вхідного зображення з ядром.

Друга частина включає операцію максимізації для певних ділянок зображення. Перший шар виділяє ознаки шляхом фільтрації ділянок вхідного зображення. Наступний шар обробляє ці ділянки, отримані після фільтрації. Це відбувається таким чином: вибирається певна область попереднього результату, знаходиться її максимальне значення, яке стає виходом для поточного шару. Цей процес повторюється для всіх ділянок зображення.

Процес згортки полягає в тому, що фільтр переміщується (ковзає) по всьому вхідному зображенню. На кожному кроці обчислюється скалярний добуток значень фільтра і відповідних значень пікселів вхідного зображення, після чого результат записується в вихідну матрицю (feature map). Результатом згортки є набір вихідних ознак (feature maps), які виділяють певні характеристики зображення, такі як контури, текстури і т.д.

Кількість фільтрів у нейронній мережі є гіперпараметром, оскільки з їх збільшенням збільшується потенційна кількість виявлених ознак і закономірностей. Проте, забагато фільтрів може спричинити виявлення нехарактерних ознак, що може призвести до перенавчання моделі. Кількість фільтрів впливає на кількість карт ознак, яка, в свою чергу, впливає на глибину вихідних даних та глибину вхідних даних для наступного шару мережі, визначаючи кількість параметрів для наступного шару. Під час навчання

фільтри навчаються розпізнавати ознаки зображення, які об'єднуються в карти ознак. Кожен фільтр створює одну карту ознак, переміщуючись по зображенню та виявляючи ознаки в сусідніх рецептивних полях. Крок переміщення (stride) може змінюватися і є одним із параметрів моделі. Хоча абсолютне положення кожної ознаки на зображенні не зберігається на карті, це неважливо, оскільки відносне положення ознак одна щодо одної залишається незмінним, зберігаючи властивості зображення. Під час навчання фільтри набувають здатності розпізнавати різні ознаки зображення, які потім об'єднуються в карти ознак. Кожен фільтр відповідає за формування однієї карти ознак. Математично операцію можна записати як (2.1).

$$m(t) = (x * k)(t) = \sum_{a=-\infty}^{a=+\infty} x(a)k(a+t), \quad (2.1)$$

Одновимірні згорткові нейронні мережі (1D CNN) знаходять широке застосування в аудіоаналізі через їхню здатність ефективно обробляти та аналізувати часові ряди, до яких належать аудіосигнали. Ці мережі мають можливість автоматично виділяти ключові ознаки з аудіоданих, такі як тональні, ритмічні та тембральні характеристики, що є критичними для завдань класифікації, розпізнавання та виявлення подій в аудіо. Завдяки цим властивостям, 1D CNN забезпечують високу точність та ефективність в обробці аудіосигналів.

Застосовуючи властивість комутативності до операції математичної згортки, можна отримати спрощену функцію (2.2), яка підходить для використання в нейронних мережах.

$$M(i,j) = (K * X)(i,j) = \sum_m \sum_n K(m,n)X(i-m,j-n), \quad (2.2)$$

де $M(i,j)$ – карта ознака з деякими координатами i та j , X – вхідне зображення, K – показник, m, n – розмірності показника.

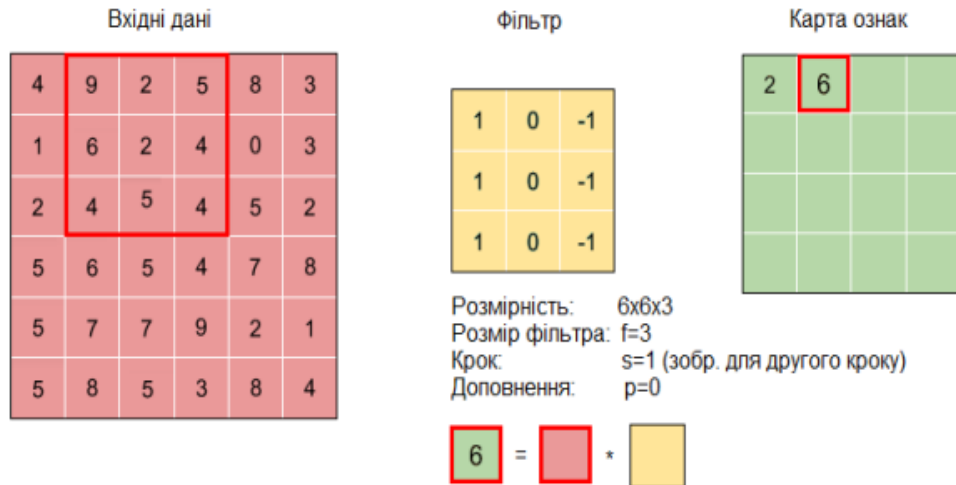


Рисунок 2.3 - Приклад одновимірного фільтра для операцій згортки 3 канали розміром зображення 6x6, 3 розміру фільтра, Розмір кроку дорівнює 1, без додавання нулів.

Кожен елемент карти ознак є результатом обчислення скалярного добутку значень з певної області вхідного зображення, яка називається рецептивним полем нейрона, з відповідними значеннями фільтра. Ознаки, які виявляють фільтри, змінюються залежно від рівня мережі, на якому вони знаходяться. На початкових рівнях фільтри можуть виявляти прості ознаки, такі як краї або градієнти кольору. На глибших рівнях фільтри здатні розпізнавати більш складні структури, такі як кути, круглі форми або навіть специфічні об'єкти.

Це досягається за рахунок того, що результати згортки на попередніх рівнях використовуються як вхідні дані для наступних рівнів. На кожному наступному рівні збільшується кількість точок початкового зображення, які впливають на кожне значення карти ознак. Завдяки цьому, чим глибше рівень, тим більше інформації враховується при обчисленні ознак, і це підвищує здатність мережі до виявлення складних ознак.

Цей підхід також забезпечує більший вплив центральних ознак на кінцевий результат. Ознаки, що знаходяться ближче до центру зображення, впливають на більшу кількість значень на карті ознак, ніж ті, що розташовані ближче до країв. Це пов'язано з тим, що центральні ознаки охоплюються більшою кількістю рецептивних полів фільтрів, що робить їх внесок у кінцевий результат більш значущим.

Фільтри обираються такими, щоб вони могли виявляти більші та рідкісні ознаки, які можуть мати важливе значення. Однак сучасні дослідження схиляються до використання менших фільтрів, оскільки вони більш ефективно використовують обчислювальні ресурси. Зі збільшенням розмірів фільтрів кількість параметрів у них зростає квадратично, що може значно ускладнити модель. Для збереження розмірів рецептивних полів, замість збільшення розміру фільтрів, часто використовується послідовне вертикальне розташування згорток. Це дозволяє збільшити глибину мережі, що сприяє кращій продуктивності, на відміну від збільшення ширини, яке потребує великих фільтрів. Інколи застосовуються навіть згортки з розміром 1, щоб

зменшити глибину даних перед виконанням обчислювально інтенсивних операцій.

Карта ознак має залежність від гіперпараметрів(2.3).

$$d_m = \frac{d_x - f + 2 * p}{s + 1}, \quad (2.3)$$

де dm – (ширина або висота), dx – розмірність вхідних даних, f – розмірність фільтрів, p – доповнення нулями, s – крок повного обходу зображення.

Після використання фільтрів і проведення потрібних операцій до отриманих результатів додається зміщення bb , яке є додатковим параметром мережі. Оскільки всі ці операції лінійні, для створення нелінійності у даних застосовується функція активації. Загальний процес перетворень у згортковому шарі можна виразити (2.4

$$y = f(w * x + b), \quad (2.4)$$

де y – вихідний результат обчислень на згортковому шарі, w – детектор ознак, x – вихідні дані, b – зсув.

2.3 . Пулінгові шари

Мета агрегувальних (пулінгових) шарів в згорткових нейронних мережах (CNN) полягає в зменшенні розмірності простору ознак, зберігаючи при цьому ключову інформацію. Це досягається шляхом об'єднання (агрегації) інформації з певних областей вхідних даних. Основні аспекти мети агрегувальних шарів включають:

- **Зменшення кількості параметрів:** Агрегувальні шари дозволяють зменшити кількість параметрів у мережі шляхом заміни кожної підгрупи ознак на одне представницьке значення. Це сприяє зменшенню обчислювальних витрат та обсягів пам'яті, необхідних для навчання та використання моделі.
- **Зменшення варіативності інформації:** Агрегація дозволяє зменшити варіативність даних, зберігаючи при цьому основні характеристики. Це допомагає уникнути перенавчання та покращує узагальнювальність моделі.
- **Стійкість до зсувів і змін:** Зменшення розмірності даних за допомогою агрегації допомагає зробити представлення більш стійким до незначних змін вхідних даних або їх позиції.
- **Збереження важливої інформації:** Незважаючи на зменшення розмірності, агрегувальні шари стараються зберегти найважливіші ознаки для подальшого аналізу та розпізнавання образів.

Отже, головна мета агрегувальних шарів полягає в покращенні ефективності та швидкодії згорткових нейронних мереж, зменшенні вимог до ресурсів та покращенні їх здатності до узагальнення.

У згорткових нейронних мережах (CNN) використовуються різні типи агрегації, або пулінгу, для зменшення розмірності даних. Основні типи агрегації включають в себе:

- **Максимальне пулінг (Max Pooling):** Для кожного підмасиву вхідних даних вибирається максимальне значення. Цей тип пулінгу допомагає виділити найбільш важливі ознаки з підмасиву.
- **Середнє пулінг (Average Pooling):** Для кожного підмасиву обчислюється середнє значення. Цей тип пулінгу дозволяє зменшити вплив шуму вхідних даних та загально згладити ознаки.
- **Мінімальне пулінг (Min Pooling):** Для кожного підмасиву вибирається мінімальне значення. Цей тип пулінгу може бути корисним для виділення найменш відображених ознак.
- **Сумісне пулінг (Sum Pooling):** Для кожного підмасиву обчислюється сума всіх значень. Цей тип пулінгу може бути корисним для деяких завдань, наприклад, у задачах регресії.
- **L2 пулінг (L2 Pooling):** Для кожного підмасиву обчислюється L2-норма (квадратний корінь з суми квадратів значень). Цей тип пулінгу може бути корисним для зменшення впливу великих значень на підсумкове значення.

Ці типи агрегації можуть застосовуватися з різними параметрами, такими як розмір ядра пулінгу та крок (стрид), для досягнення певних ефектів у зменшенні розмірності та збереженні важливої інформації.

Параметри агрегаційного (пулінгового) шару в згорткових нейронних мережах включають розмір ядра (kernel size), яке визначає розмірність області, що обробляється; стрид (stride), який вказує, на скільки пікселів зсувається ядро пулінгу кожного кроку; тип пулінгу (pooling type), такий як максимальне, середнє або мінімальне значення з підмасиву; заповнення (padding), яке визначає, як вирівнюються розміри вихідних даних; та можливі додаткові параметри, такі як розмір пулінгу по кожному виміру вхідних даних або кілька шарів пулінгу послідовно. Ці параметри дозволяють підлаштувати модель для різних завдань та вхідних даних.

Стрид (stride) - це параметр в агрегаційних (пулінгових) шарах згорткових нейронних мереж, що вказує, на скільки пікселів або одиниць відбувається зсув (крок) ядра пулінгу кожного разу при обробці вхідних даних.

Наприклад, якщо вхідні дані мають розмірність 5x5 пікселів, а ядро пулінгу має розмірність 2x2 і стрид 2, то перше використання пулінгу зменшить розмірність до 2x2, оскільки ядро буде переміщуватись на 2 пікселя вниз і вправо.

Тобто, стрид впливає на розмірність вихідних даних після застосування пулінгу: більший стрид призводить до зменшення розмірності вихідних даних, тоді як менший дозволяє зберегти більше деталей за рахунок більшого перекриття областей під час обробки.

Агрегаційні шари поділяються на два основні типи: усереднювальні (average) та максимальні (maximal). Іноді також використовуються мінімальні (minimal) або агрегаційні шари з використанням L2-норми. Усереднювальні шари обчислюють середнє значення для кожного елемента з відповідної підрозділу нейронів з минулого шару, тоді як певні шари визначають максимальне значення. Формально обидва типи агрегації для для максимізаційного(2.5) та мінімального(2.6) агрегування.)

:

$$p(i, j) = \frac{\sum_{m, n} x(i - m, j - n)}{m * n}, \quad (2.5)$$

$$p(i, j) = \max_{i, j}(x(i - m, j - n)), \quad (2.6)$$

де $p(i, j)$ – значення елемента поточного рівня з координатами i та j , x – вхідні дані з попередніх рівнів, m, n – розмірність рецептивного поля.

Попередній шар отримує ознаки шляхом фільтрації певної частини вхідного зображення. Наступний шар обробляє частини зображення, що були отримані після фільтрації. Це відбувається так: для кожної частини попереднього результату вибирається ділянка із максимальним значенням, яке стає виходом поточного шару. Цей процес повторюється для кожної частини. Операція виглядає наступним чином:

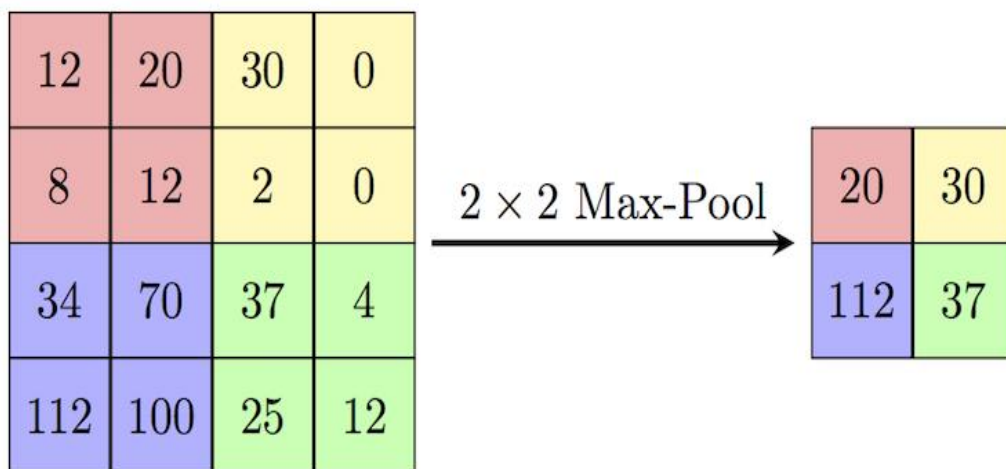


Рисунок 2.4 - Процес пулінгу з фільтром 2×2 та з кроком два

Агрегувальні шари в згорткових нейронних мережах (CNN) відіграють важливу роль у покращенні ефективності та швидкодії моделей. Вони допомагають зменшити розмірність даних, зберігаючи при цьому ключову інформацію. Основні типи агрегації включають максимальне, середнє та мінімальне пулінги, які вибирають відповідно максимальне, середнє або мінімальне значення з підмасиву. Параметри агрегаційного шару, такі як розмір ядра, стрид і тип пулінгу, можуть бути налаштовані для досягнення певних ефектів у зменшенні розмірності та збереженні важливої інформації. Використання агрегувальних шарів дозволяє покращити швидкість та ефективність нейронних мереж, знижуючи вимоги до ресурсів та поліпшуючи їх здатність до узагальнення.

2.4 . Функції втрат

Вибір та налаштування функції втрат є ключовим аспектом у створенні нейронної мережі, оскільки це суттєво впливає на якість і швидкість навчання моделі, а також на її здатність точно оцінювати результати. Значення функції втрат є критерієм ефективності роботи мережі, оскільки вона дозволяє об'єднати всі переваги, недоліки та помилки мережі в одне значення. Це дозволяє характеризувати мережу таким чином, щоб можна було порівняти результати роботи для різних варіантів моделі за точністю. Крім того, функція втрат повинна враховувати особливості конкретної задачі, для якої використовується мережа.

Функція втрат (або функція втрати) визначає, як саме вимірюється рівень втрат. Вибір правильної функції втрати залежить від конкретної задачі мережі. Ось деякі типові функції втрат для різних типів задач:

- **Класифікація:** Для задачі класифікації часто використовується категоріальна крос-ентропія (categorical cross-entropy), особливо для багатокласової класифікації. Ця функція штрафувє модель за неправильні передбачення класів.
- **Регресія:** У задачах регресії може використовуватися середньоквадратична помилка (mean squared error, MSE), яка вимірює середньоквадратичну різницю між прогнозованими та фактичними значеннями.
- **Семантична сегментація:** Для сегментації зображень може застосовуватися категоріальна крос-ентропія на рівні пікселів або додаткові функції втрат для врахування структурної схожості між сусідніми пікселями.
- **Генеративні моделі:** У генеративних моделях, таких як генеративно-змістовні мережі (GAN), можуть використовуватися спеціальні функції втрат, такі як бінарна крос-ентропія для вимірювання точності генератора та дискримінатора.

Важливо вибирати функцію втрат, яка відповідає конкретній задачі та вимогам моделі, оскільки це може суттєво вплинути на якість та швидкість навчання.

					КНУ.РБ.123.24.06.ЗНМС	Арк.
Арк.	№ документа	Підпис	Дата			

Функція перехресної ентропії (cross-entropy) використовується в нейронних мережах як функція втрат для задач класифікації. Вона вимірює відстань між двома ймовірнісними розподілами: фактичним розподілом класів і передбачуваним розподілом, який генерується моделлю. Ця функція допомагає моделі оцінити, наскільки добре вона класифікує дані. Функція (2.7) штрафує модель за неправильні класифікації, збільшуючи втрати, коли відстань між фактичним і передбачуваним розподілом збільшується. Використання функції перехресної ентропії допомагає навчити модель класифікувати дані з більшою точністю.

$$L(p, y) = - \sum_n y_n \ln p_n, n \in [1, N], \quad (2.7)$$

де p – належність даних до вибраних класів, y – справжні значення для вхідних даних, N – кількість класів.

2.5 . Активаційні функції

Активаційні функції є ключовими компонентами у штучних нейронних мережах, вони визначають вихідне значення нейрона залежно від вхідних даних і ваг. Вони використовуються для надання неелінійності та можливості моделі апроксимувати складні залежності в даних. Ось деякі з найпоширеніших активаційних функцій:

Сигмоїдальна функція Використовується для бінарної класифікації або вихідного шару у мережах, де потрібно вивести значення у діапазоні від 0 до 1.

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad (2.8)$$

Гіперболічний тангенс (tanh): Функція подібна до сигмоїдальної, але видає значення у діапазоні від -1 до 1, що може поліпшити швидкість навчання.

$$(f(x) = \tanh(x)) \quad (2.9)$$

ReLU (Rectified Linear Activation): Ця функція нулює від'ємні значення і залишає позитивні без змін. Вона допомагає уникнути проблеми з втратою градієнту і є однією з найпоширеніших активаційних функцій для прихованих шарів.

$$y = f(x) = \max(0, x), \quad (2.10)$$

де y – елемент поточного рівня, x – елемент з вхідних даних

Softmax: Ця функція використовується для отримання ймовірностного розподілу на виході мережі. Вона приймає вектор значень та повертає вектор, де кожне значення виражає ймовірність належності до певного класу.

$$y_i = f(x)_i = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}}, \forall i, j \in (1, N), \quad (2.11)$$

де y_i – елемент шару з індексом i , x_i – елемент попереднього шару з індексом i , N – кількість елементів на попередньому шарі.

Ці активаційні функції грають важливу роль у навчанні нейронних мереж і впливають на їхню здатність апроксимувати складні залежності у даних. Вибір конкретної функції залежить від конкретної задачі та особливостей даних.

Висновок до другого розділу

Convolutional Neural Networks (CNNs) є потужним інструментом у глибинному навчанні для обробки зображень і відео. Вони використовуються для розв'язання різноманітних задач, включаючи класифікацію зображень, семантичну сегментацію, визначення об'єктів та багато інших. Основні компоненти CNN включають згорткові шари для виявлення різних функцій та областей у зображеннях, агрегаційні шари для зменшення розмірності та підсумовування інформації, а також повнозв'язані шари для класифікації.

Використання відповідної архітектури та належного налаштування гіперпараметрів, таких як розмір фільтрів, кількість шарів і розмір пакета для тренування, може значно покращити продуктивність мережі. Функції втрати та активаційні функції важливі для ефективного навчання моделі та досягнення високої точності.

Загалом, CNN є потужним інструментом для роботи з зображеннями та відео, і їх застосування може призвести до вражаючих результатів у багатьох галузях, від медичного зображення до автомобільних технологій.

РОЗДІЛ 3 ПОБУДОВА МОДЕЛІ ДЛЯ ЧИТАННЯ ТЕКСТУ

У цьому розділі ми розглянемо, як створити модель для розпізнавання тексту, використовуючи мову програмування Python, бібліотеки OpenCV та Keras, а також зглибимося в основи згорткових нейронних мереж (CNN). Ми почнемо з основних понять і поступово перейдемо до побудови моделі, яка зможе ефективно розпізнавати текст на зображеннях. Розглянемо кроки підготовки даних, побудови архітектури нейронної мережі, навчання моделі, а також її тестування та оцінювання.

3.1. Підготовка зображення

Першим кроком у розпізнаванні тексту є розділення його на окремі літери. Для цього використовуємо бібліотеку OpenCV, зокрема функцію `findContours`. Використовуємо функцію `cv2.imread`, щоб завантажити зображення з текстом у кольоровому форматі. Перетворюємо кольорове зображення у відтінки сірого за допомогою `cv2.cvtColor`. Потім застосовуємо порогову обробку `cv2.threshold`, щоб перетворити сіре зображення у двокольорове (чорно-біле). Застосовуємо ерозію за допомогою `cv2.erode`, щоб злегка збільшити контури тексту на зображенні. Це допомагає зменшити шуми та покращити якість подальшого визначення контурів.

OpenCV (Open Source Computer Vision Library) - це бібліотека комп'ютерного зору та машинного навчання з відкритим вихідним кодом, розроблена для роботи з зображеннями і відео. OpenCV надає набір інструментів для обробки зображень, аналізу, відстеження об'єктів, розпізнавання образів та багато іншого.

					КНУ.РБ.123.24.06.ПМДЧТ		
Змн.	Арк.	№ документа	Підпис	Дата			
Розробив		Грязнов			Літера	Аркуш	Аркушів
Перевірив							
Н.контроль		Кузнецов			КІ-20		
Затвердив		Купін					
					ПОБУДОВА МОДЕЛІ ДЛЯ ЧИТАННЯ ТЕКСТУ		

Основні можливості OpenCV включають:

- Завантаження, збереження та відображення зображень та відео.
- Обробка та аналіз зображень, такі як фільтрація, морфологічна обробка, виявлення країв, сегментація та інші операції.
- Відстеження руху об'єктів та розпізнавання облич, рук, об'єктів та тексту.
- Взаємодія з камерами та відеопотоками для захоплення та обробки відео.
- Робота з 3D-зображеннями та обробка відео у реальному часі.

OpenCV широко використовується у мовах програмування, таких як C, C++, Python та Java. Обробка зображень - це процес застосування різних операцій до зображення для покращення якості або витягнення корисної інформації.

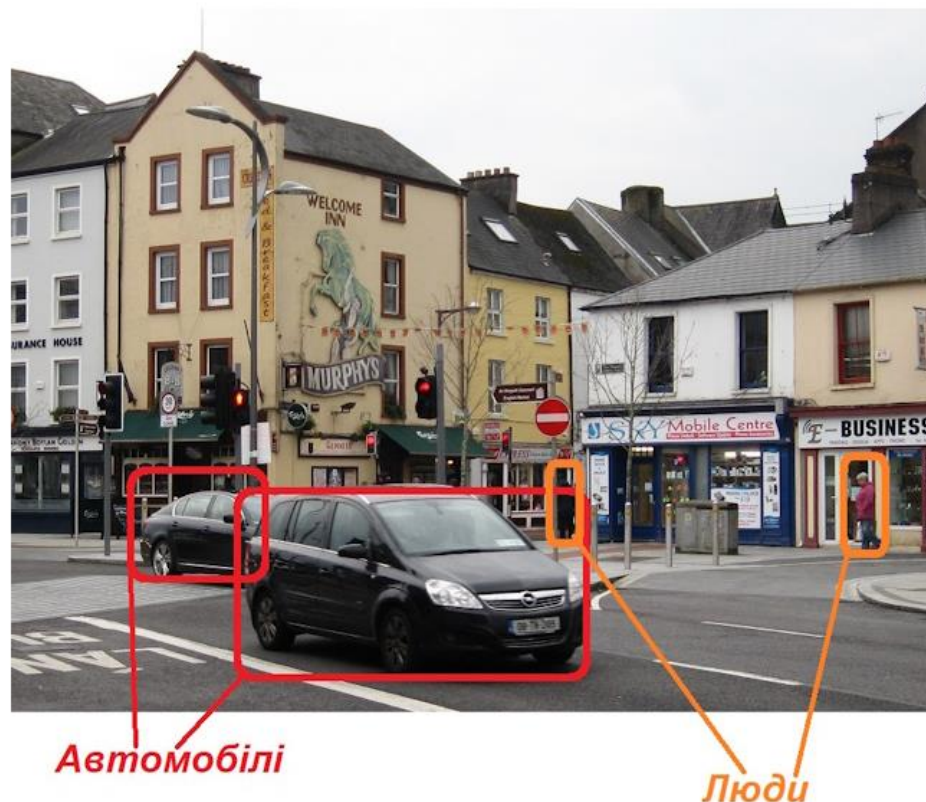


Рисунок 3.1 - Приклад виявлення об'єктів комп'ютерним зором

Основні етапи обробки зображень включають імпорт зображення, аналіз та обробку, а також вивід результату (зміненого зображення або звіту). У сучасному світі OpenCV дуже популярна для мов програмування Python завдяки його зручному синтаксису та багатому набору вбудованих модулів. Крім того, OpenCV легко поєднується з іншими модулями Python, такими як NumPy, matplotlib, tensorflow та інші.

Спочатку зчитується кольорове зображення та перетворюється у відтінки сірого, після чого виконується порогове перетворення для отримання чорно-білого зображення. Після цього застосовується ерозія для покращення контурів. Далі використовується функція `cv2.findContours` для знаходження

контурів на зображенні, після чого відображаються області, визначені контурами, на оригінальному зображенні.

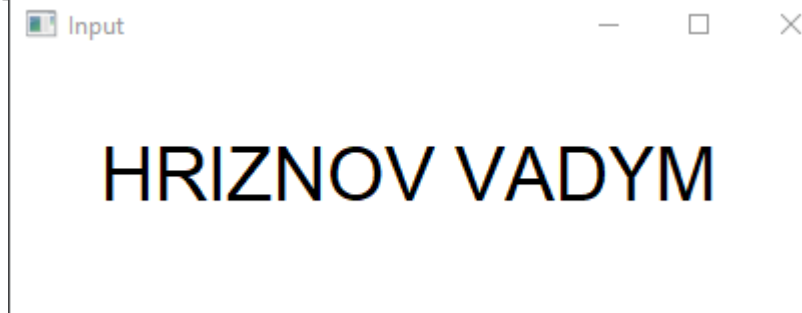


Рисунок 3.2 - Загальний контур



Рисунок. 3.3. - Контур літер



Рисунок 3.5. - Внутрішній контур

3.2. MNIST

Наступний крок полягає в збереженні кожної літери, відмасштабованої до розміру квадрату 28x28 пікселів (це стандартний формат для бази даних MNIST). Завдяки тому, що OpenCV побудований на основі numpy, ми можемо використовувати функції роботи з масивами для обрізки та масштабування зображень.

MNIST – це база даних, яка широко використовується в області машинного навчання, особливо для тестування та навчання алгоритмів розпізнавання образів. Ось кілька основних характеристик цієї бази даних:

Основні характеристики MNIST:

1. Зображення цифр: MNIST містить зображення рукописних цифр від 0 до 9.

2. Розмір зображень: Кожне зображення має розмір 28x28 пікселів, що дає 784 пікселі в загальному.

Кількість зображень:

					КНУ.РБ.123.24.06.ПМДЧТ	Арк.
Арк.	№ документа	Підпис	Дата			

- Навчальна вибірка: 60,000 зображень.
- Тестова вибірка: 10,000 зображень.

3. Формат даних: Зображення збережені в відтинках сірого, де кожен піксель представлений значенням від 0 (чорний) до 255 (білий).

4. Масштабовані та центровані зображення: Всі зображення вже обрізані та центровані в рамках 28x28 пікселів, що спрощує процес обробки.



Рисунок. 3.5 - Розбиття на літери

Використання в машинному навчанні:

MNIST є ідеальною базою даних для тестування нових алгоритмів розпізнавання образів, відлагодження моделей нейронних мереж та порівняння продуктивності різних алгоритмів



Рисунок 3.6 - Приклади зображень в базі MNIST

Вихідний датасет EMNIST має 62 різних символи (A..Z, 0..9 та ін):
 emnist_labels = [48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122]

Нейронна мережа має 62 виходи і отримує на вході зображення розміром 28x28. Після розпізнавання, наприклад, число "1" буде мати відповідний вихід у мережі.

Далі визначаємо модель нейронної мережі для розпізнавання символів EMNIST (розширений алфавітний набір) поетапно.

Модель створюється як послідовна модель, де шари додаються один за одним. Додаються два шари згорткових нейронів (Convolution2D), що допомагають виявляти важливі ознаки у зображеннях. Перший шар має 32 фільтри з розміром ядра 3x3, а другий шар має 64 фільтри того ж розміру. Додається шар пулінгу (MaxPooling2D), який зменшує розмірність зображення шляхом обрання максимального значення у кожному пікселі з певної області. Додається шар випадкового вимикання (Dropout), що

допомагає уникнути перенавчання шляхом випадкового вимикання деяких нейронів під час навчання. Додається шар Flatten, який перетворює вихід попередніх шарів в одномірний вектор для передачі його до повнозв'язного шару. Додаються два повнозв'язних шари (Dense). Перший має 512 нейронів з функцією активації ReLU, а другий має кількість виходів, що дорівнює кількості класів ($\text{len}(\text{emnist_labels})$) з функцією активації softmax для розпізнавання класів. Модель компілюється з використанням функції втрат категоріальної кросс-ентропії, оптимізатора Adadelta і метрик точності. Функція повертає зібрану модель готову до навчання.

3.3. Навчання моделі

Далі, відбувається підготовка даних для навчання моделі нейронної мережі для розпізнавання символів EMNIST. Використовуючи бібліотеку `idx2numpy`, дані завантажуються з файлів у форматі IDX у масиви `X_train`, `y_train`, `X_test` та `y_test`. Зображення перетворюються з формату (кількість зображень, 784) у формат (кількість зображень, 28, 28, 1), де остання ось відповідає за кольори, у нашому випадку один канал, оскільки це чорно-білі зображення. Для швидшого навчання моделі вибирається лише частина даних. У нас $k = 10$, тому використовується лише кожна 10-та зображення. Зображення нормалізуються до значень від 0 до 1, ділячи кожне значення на 255, що є максимальним значенням пікселя. Категоріальні мітки перетворюються в формат one-hot з використанням `keras.utils.to_categorical`.

Початкові фільтри в нейронних мережах зазвичай існують для виявлення базових характеристик, таких як лінії або прості текстури. Пізніші фільтри у глибоких шарах мережі стають більш складними і комбінують виявлені раніше базові характеристики для розпізнавання складніших шаблонів. Наприклад, якщо перший рівень виявляє краї, то глибший рівень може використовувати цю інформацію для визначення форм.

Мережа CNN зазвичай також включає шари зменшення просторових розмірів зображень між цими фільтрами. Це допомагає підвищити ефективність і точність мережі. На останніх рівнях модель приймає остаточне рішення, наприклад, класифікує об'єкти на зображенні, використовуючи результати попередніх шарів.

```
Epoch 1/30
1091/1091 — 142s 127ms/step - accuracy: 0.0448 - loss: 4.0842 - val_accuracy: 0.0971 - val_loss: 3.9428 - learning_rate: 0.0010
Epoch 2/30
1091/1091 — 144s 132ms/step - accuracy: 0.1005 - loss: 3.8882 - val_accuracy: 0.1753 - val_loss: 3.6618 - learning_rate: 0.0010
Epoch 3/30
1091/1091 — 142s 130ms/step - accuracy: 0.1403 - loss: 3.6380 - val_accuracy: 0.2973 - val_loss: 3.4127 - learning_rate: 0.0010
Epoch 4/30
1091/1091 — 142s 130ms/step - accuracy: 0.2162 - loss: 3.4236 - val_accuracy: 0.3944 - val_loss: 3.1773 - learning_rate: 0.0010
Epoch 5/30
744/1091 — 43s 124ms/step - accuracy: 0.2886 - loss: 3.2387
```

Рисунок 3.7 - Процес навчання системи

Сам процес зайняв близько півтори години. Час повністю залежить від ваших комплектуючих, тому на різних ПК він може відрізнятись. □ Модель навчається на тренувальній вибірці протягом кількох епох. Під час кожної епохи модель оновлює свої ваги на основі зворотного поширення помилки, щоб мінімізувати функцію втрат. Валідаційна вибірка використовується для

перевірки моделі після кожної епохи, що дозволяє оцінити її здатність до узагальнення на нових даних.

```

Трив: 17:08:17 - втрачено: 0:20:27 - loss: 1.2189 - val_loss: 0.9487 - val_acc: 0.7012 - result_dir: 0:00:00
Епохи: 30/30
  
```

Рис. 3.8 Закінчення навчання системи

Далі завантажуюмо збережену модель нейронної мережі з файлу `emnist_letters.h5` та використовуємо її для передбачення символу на зображенні. Функція `emnist_predict_img` нормалізує та обробляє вхідне зображення, а потім використовує модель для передбачення класу символу. Результат передбачення повертається у вигляді символу ASCII, відповідного передбаченому класу.

```

1/1 ██████████ 0s 500ms/step
1/1 ██████████ 0s 40ms/step
1/1 ██████████ 0s 45ms/step
1/1 ██████████ 0s 49ms/step
1/1 ██████████ 0s 41ms/step
1/1 ██████████ 0s 48ms/step
1/1 ██████████ 0s 45ms/step
1/1 ██████████ 0s 55ms/step
1/1 ██████████ 0s 56ms/step
1/1 ██████████ 0s 63ms/step
1/1 ██████████ 0s 52ms/step
1/1 ██████████ 0s 56ms/step
VADIM HRIAZNOV
  
```

Рисунок 3.9 - Результат

3.4 Оцінка моделі

Точність вимірює відсоток правильно класифікованих символів. Це основна метрика для задачі класифікації. Точність обчислюється як відношення кількості правильно передбачених прикладів до загальної кількості прикладів.

Таблиця 3.1 Оцінка кожної 3 епохи

Epoch	Accuracy	Loss	Val_Accuracy	Val_Loss	Learning Rate
1	0.0350	4.0990	0.1376	3.9425	0.0010
3	0.1745	3.6240	0.3438	3.3565	0.0010
6	0.3500	2.9158	0.4613	2.5635	0.0010
9	0.4397	2.3802	0.5182	2.0621	0.0010
12	0.4875	2.1182	0.5589	1.8066	0.0010
15	0.5165	1.9446	0.5882	1.6485	0.0010
18	0.5404	1.8139	0.6059	1.5378	0.0010
21	0.5589	1.7259	0.6206	1.4538	0.0010
24	0.5643	1.6808	0.6333	1.3857	0.0010
27	0.5801	1.6064	0.6427	1.3311	0.0010
30	0.5872	1.5686	0.6520	1.2870	0.0010

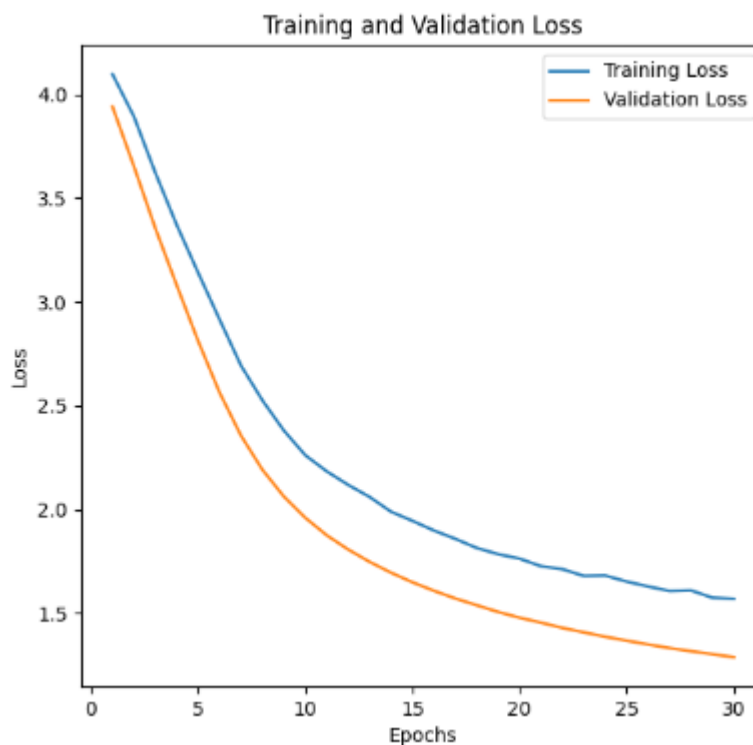


Рисунок 3.10 - Графік втрат

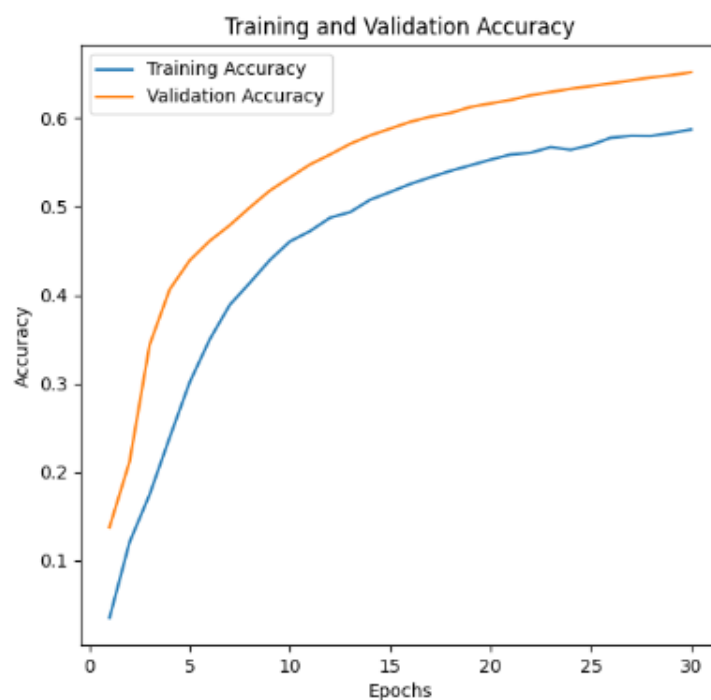


Рисунок 3.11 - Графік точності

Детально аналізуємо результати навчання моделі на основі наданих даних, зокрема звернемо увагу на точність та втрати моделі протягом 30 епох. Точність на тренувальній та валідаційній вибірках зростала стабільно протягом усіх 30 епох, що свідчить про те, що модель поступово навчалась краще розпізнавати патерни в даних. Зменшення втрат: Втрати на тренувальній та валідаційній вибірках постійно зменшувались, що вказує на те, що модель ставала менш помилковою.

Валідаційна точність на останній епосі досягла 65.20%, що є досить хорошим результатом для задачі розпізнавання тексту.

На першій епосі точність моделі була на рівні 0.0350 для тренувального набору та 0.1376 для валідаційного набору, при цьому втрати склали 4.0990 та 3.9425 відповідно. Протягом наступних епох ми спостерігали стабільне покращення показників точності та зменшення втрат.

До третьої епохи точність тренувального набору досягла 0.1745, а валідаційного - 0.3438, що супроводжувалося відповідним зменшенням втрат до 3.6240 та 3.3565. Таке покращення можна пояснити поступовим коригуванням ваг мережі під час навчання.

Найбільше покращення відбулося між четвертою та п'ятою епохами, коли точність валідаційного набору збільшилася з 0.4064 до 0.4392, а втрати зменшилися з 3.0810 до 2.8119. Це свідчить про те, що модель успішно навчалася на основі наданих даних.

До кінця навчання, на тридцятій епосі, модель досягла точності 0.5872 для тренувального набору та 0.6520 для валідаційного, з втратами 1.5686 та 1.2870 відповідно. Це свідчить про значне покращення продуктивності моделі протягом навчання.

Таблиця 3.2 Основні шари та їх параметри

Крок	Опис	Тип шару	Кількість фільтрів	Розмір фільтрів	Функція активації	Інші параметри
1	Введення зображення	Вхідний шар	-	28x28	-	-
2	Згортка	Convolution2D	32	3x3	ReLU	-
3	Згортка	Convolution2D	64	3x3	ReLU	-
4	Пулінг	MaxPooling2D	-	2x2	-	-
5	Випадкове вимикання	Dropout	-	-	-	Рівень drop: 0.25
6	Перетворення вектора	Flatten	-	-	-	-
7	Повнозв'язний шар	Dense	512	-	ReLU	-
8	Повнозв'язний шар	Dense	62 (кількість класів)	-	Softmax	-
9	Компіляція моделі	-	-	-	-	Втрати: категоріальна

Розроблена модель CNN для розпізнавання тексту показує хороші результати, поступово покращуючи точність та зменшуючи втрати на навчальній та валідаційній вибірках. Використання різних методів оптимізації та покращення архітектури може ще більше підвищити ефективність моделі. Застосування удосконалень дозволить досягти більш високих результатів та забезпечити стабільне розпізнавання тексту на зображеннях. Таблиця ілюструє основні кроки та параметри побудови моделі згорткової нейронної мережі для розпізнавання символів з використанням бібліотек Python, OpenCV, та Keras.

					КНУ.РБ.123.24.06.ПМДЧТ	Арк.
	Арк.	№ документа	Підпис	Дата		

ВИСНОВОК

Загальний висновок цієї дипломної роботи полягає в дослідженні та вдосконаленні методів розпізнавання простих об'єктів на двовимірних монохромних зображеннях з використанням сучасних обчислювальних підходів, таких як згорткові нейронні мережі (CNN). Протягом роботи були досягнуті значні результати, які можуть слугувати основою для подальших досліджень та практичного застосування в різних сферах.

Робота містить детальний огляд існуючих методів обробки зображень, включаючи класичні методи, такі як фільтрація, гістограмне вирівнювання та морфологічні операції. Було проаналізовано сучасні методи машинного навчання, зокрема згорткові нейронні мережі, які показали високу ефективність у задачах розпізнавання об'єктів. Показано, що CNN є надзвичайно ефективними для задач розпізнавання об'єктів завдяки їх здатності автоматично витягувати релевантні ознаки зображень. Розроблені та протестовані моделі CNN продемонстрували високу точність розпізнавання об'єктів, що підтверджує їх практичну цінність.

Практична частина роботи включала розробку та реалізацію програмних прототипів для задачі розпізнавання об'єктів. Проведена експериментальна перевірка ефективності розроблених моделей на реальних даних підтвердила їх високу точність та надійність. Водночас було виконано порівняння класичних та сучасних методів обробки зображень, що дозволило визначити оптимальні підходи для різних типів задач. Було виявлено ключові переваги використання CNN, такі як здатність до автоматичного навчання ознак та висока точність, а також можливі напрямки для подальшого вдосконалення.

					КНУ.РБ.123.24.06.В			
Змн.	Арк.	№ документа	Підпис	Дата				
Розробив	Грязнов				ВИСНОВОК	Літера	Аркуш	Аркушів
Перевірив								
Н.контроль	Кузнецов				КІ-20			
Затвердив	Купін							

Подальші дослідження можуть включати оптимізацію та вдосконалення алгоритмів для підвищення їх ефективності в умовах обмежених обчислювальних ресурсів, дослідження можливості використання новітніх архітектур нейронних мереж для покращення продуктивності та точності, а також застосування розроблених методів у нових сферах, таких як розпізнавання жестів, аналіз медичних зображень, автоматизація виробничих процесів та системи безпеки. Також варто розглянути можливість розробки ефективних рішень для мобільних пристроїв та периферійних обчислювальних платформ, що забезпечить високу продуктивність в реальному часі.

Отже, дипломна робота показала, що використання сучасних методів глибокого навчання, зокрема згорткових нейронних мереж, значно покращує ефективність та точність розпізнавання об'єктів на двовимірних монохромних зображеннях. Отримані результати підтвердили доцільність застосування цих методів у різних практичних задачах. Подальший розвиток та вдосконалення запропонованих підходів може сприяти значному прогресу в області комп'ютерного зору та машинного навчання, що відкриває нові можливості для автоматизації та підвищення ефективності різних галузей.

					КНУ.РБ.123.24.06.В	Арк.
	Арк.	№ документа	Підпис	Дата		

ПЕРЕЛІК ПОСИЛАНЬ

1. Гудз, О. І. Основи нейронних мереж / О. І. Гудз. – Київ : Наукова думка, 2018. – 240 с.
2. Сітков, О. В. Комп'ютерний зір та його застосування / О. В. Сітков, М. П. Гребенюк. – Харків : ХНУРЕ, 2019. – 310 с.
3. Ivakhnenko, A. G. Heuristic self-organization in problems of engineering cybernetics / A. G. Ivakhnenko, V. G. Lapa. – New York : Taylor & Francis, 2016. – 362 p.
4. Nielsen, M. Neural Networks and Deep Learning URL: <http://neuralnetworksanddeeplearning.com/>.(дата звернення: 11.03.2024).
5. Brownlee, J. Deep Learning for Computer Vision URL: <https://machinelearningmastery.com/deep-learning-for-computer-vision/>.(дата звернення: 27.03.2024).
6. Chollet, F. Keras Documentation URL: <https://keras.io/>.(дата звернення: 09.24.2024).
7. LeCun, Y. Gradient-based learning applied to document recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // Proceedings of the IEEE. – 1998. – Vol. 86, No. 11. – P. 2278-2324.
8. Krizhevsky, A. ImageNet Classification with Deep Convolutional Neural Networks / A. Krizhevsky, I. Sutskever, G. E. Hinton // Advances in Neural Information Processing Systems. – 2012. – P. 1097-1105.
9. Szegedy, C. Going deeper with convolutions / C. Szegedy, W. Liu, Y. Jia, et al. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2015. – P. 1-9.

					КНУ.РБ.123.24.06.ІІІ		
Змн.	Арк.	№ документа	Підпис	Дата			
Розробив	Грязнов				Літера	Аркуш	Аркушів
Перевірив							
Н.контроль	Кузнецов				ПЕРЕЛІК ПОСИЛАНЬ KI-20		
Затвердив	Купін						

10. Goodfellow, I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. – Cambridge, MA : MIT Press, 2016. – 775 p.
11. Bishop, C. M. Pattern Recognition and Machine Learning / C. M. Bishop. – New York : Springer, 2006. – 738 p.
12. Chollet, F. Deep Learning with Python / F. Chollet. – Shelter Island, NY : Manning Publications, 2018. – 384 p.
13. He, K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P. 770-778.
14. OpenCV Documentation – URL: <https://docs.opencv.org/>. (дата звернення: 02.05.2024).
15. Coursera. Deep Learning Specialization URL: <https://www.coursera.org/specializations/deep-learning>. (дата звернення: 05.04.2024).
16. Антонов О.М. Сучасні методи обробки зображень. – Київ: Наукова думка, 2023. – 240 с.
17. Бойко Л.І., Кравець Д.С. Застосування штучного інтелекту в розпізнаванні образів. – Львів: Вид. дім "Світ", 2022. – 220 с.
18. Грищенко Н.В., Сидоренко П.В. Основи цифрової обробки зображень. – Одеса: Одеський нац. університет, 2023. – 230 с.
19. Дерев'янко С.П., Лисенко І.В. Нейронні мережі в комп'ютерному зорі. – Вінниця: ВНТУ, 2023. – 260 с.
20. Жуковський М.О., Козак Л.В. Згорткові нейронні мережі для аналізу зображень. – Дніпро: Вид. дім ДНУ, 2022. – 250 с.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```

import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

import numpy as np
import pathlib
import imghdr
from typing import *
from matplotlib import pyplot as plt
from scipy import io as spio
import idx2numpy
import tensorflow as tf
from keras import backend as K
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense,
BatchNormalization
from keras.optimizers import SGD, RMSprop, Adam
from tensorflow import keras
# Function to print the digit in a specific format
def print_digit(d):
    print(d.shape)
    for x in range(28):
        s = ""
        for y in range(28):
            s += "{0:.1f} ".format(d[28 * y + x])
        print(s)

# Function to print the digit in a 2D array format
def print_digit_2d(d):
    print(d.shape)
    for y in range(d.shape[0]):
        s = ""
        for x in range(d.shape[1]):
            s += "{0:.1f} ".format(d[x][y])
        print(s)

# Define EMNIST labels
emnist_labels = [i for i in range(48, 58)] + [i for i in range(65, 91)] + [i for i
in range(97, 123)]

# Function to build the EMNIST model
def build_emnist_model():
    model = Sequential([

```

```

        Conv2D(32, (3, 3), padding='valid', input_shape=(28, 28, 1),
activation='relu'),
        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),
        Dropout(0.25),
        Flatten(),
        Dense(512, activation='relu'),
        Dropout(0.5),
        Dense(len(emnist_labels), activation='softmax')
    ])
    model.compile(loss='categorical_crossentropy', optimizer='adadelta',
metrics=['accuracy'])
    return model

# Function to train the EMNIST model
def train_emnist_model(model):
    start_time = time.time()

    data_path = 'D:\\Temp\\1\\'
    X_train = idx2numpy.convert_from_file(data_path + 'emnist-byclass-
train-images-idx3-ubyte')
    y_train = idx2numpy.convert_from_file(data_path + 'emnist-byclass-
train-labels-idx1-ubyte')
    X_test = idx2numpy.convert_from_file(data_path + 'emnist-byclass-test-
images-idx3-ubyte')
    y_test = idx2numpy.convert_from_file(data_path + 'emnist-byclass-test-
labels-idx1-ubyte')

    X_train = np.reshape(X_train, (X_train.shape[0], 28, 28, 1))
    X_test = np.reshape(X_test, (X_test.shape[0], 28, 28, 1))

    print(X_train.shape, y_train.shape, X_test.shape, y_test.shape,
len(emnist_labels))

# Reduce dataset size for testing
X_train = X_train[:len(X_train) // 10]
y_train = y_train[:len(y_train) // 10]
X_test = X_test[:len(X_test) // 10]
y_test = y_test[:len(y_test) // 10]

# Normalize dataset
X_train = X_train.astype(np.float32) / 255.0
X_test = X_test.astype(np.float32) / 255.0

y_train_cat = keras.utils.to_categorical(y_train, len(emnist_labels))
y_test_cat = keras.utils.to_categorical(y_test, len(emnist_labels))

```



```

# Set learning rate reduction
lr_reduction =
keras.callbacks.ReduceLROnPlateau(monitor='val_accuracy', patience=3,
verbose=1, factor=0.5, min_lr=0.00001)

model.fit(X_train, y_train_cat, validation_data=(X_test, y_test_cat),
callbacks=[lr_reduction], batch_size=64, epochs=30)
print("Training done, duration:", time.time() - start_time)

# Function to make predictions with the EMNIST model
def predict_with_emnist(model, image_file):
    img = keras.preprocessing.image.load_img(image_file, target_size=(28,
28), color_mode='grayscale')
    return predict_image(model, img)

# Function to predict the label of an image
def predict_image(model, img):
    img_arr = np.expand_dims(img, axis=0)
    img_arr = 1 - img_arr / 255.0
    img_arr[0] = np.rot90(img_arr[0], 3)
    img_arr[0] = np.fliplr(img_arr[0])
    img_arr = img_arr.reshape((1, 28, 28, 1))

    prediction = model.predict([img_arr])
    result = np.argmax(prediction, axis=1)
    return chr(emnist_labels[result[0]])

# Function to extract letters from an image
def extract_letters(image_file, output_size=28):
    img = cv2.imread(image_file)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    _, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY)
    img_erode = cv2.erode(thresh, np.ones((3, 3), np.uint8), iterations=1)

    contours, hierarchy = cv2.findContours(img_erode, cv2.RETR_TREE,
cv2.CHAIN_APPROX_NONE)

    output = img.copy()
    letters = []
    for idx, contour in enumerate(contours):
        x, y, w, h = cv2.boundingRect(contour)
        if hierarchy[0][idx][3] == 0:
            cv2.rectangle(output, (x, y), (x + w, y + h), (70, 0, 0), 1)
            letter_crop = gray[y:y + h, x:x + w]

```

```

# Resize letter canvas to square
size_max = max(w, h)
letter_square = 255 * np.ones((size_max, size_max), dtype=np.uint8)
if w > h:
    y_pos = size_max // 2 - h // 2
    letter_square[y_pos:y_pos + h, 0:w] = letter_crop
elif w < h:
    x_pos = size_max // 2 - w // 2
    letter_square[0:h, x_pos:x_pos + w] = letter_crop
else:
    letter_square = letter_crop

    letters.append((x, w, cv2.resize(letter_square, (output_size,
output_size), interpolation=cv2.INTER_AREA)))

    letters.sort(key=lambda x: x[0], reverse=False)
    return letters

# Function to convert an image to string
def image_to_string(model, image_file):
    letters = extract_letters(image_file)
    output_str = ""
    for i in range(len(letters)):
        dn = letters[i + 1][0] - letters[i][0] - letters[i][1] if i < len(letters) - 1
else 0
        output_str += predict_image(model, letters[i][2])
        if dn > letters[i][1] / 4:
            output_str += ' '
    return output_str

if __name__ == "__main__":
    # Uncomment the following lines to train the model
    # model = build_emnist_model()
    # train_emnist_model(model)
    # model.save('emnist_letters.h5')

    model = keras.models.load_model('emnist_letters.h5')
    result = image_to_string(model, "text1.png")
    print(result)

```