СЕКЦІЯ «ІНФОРМАТИКА»

УДК 004.032.26

DOI 10.31651/2076-5886-2019-2-59-72

PACS 07.05.Mh, 07.05.Kf, 07.05.Pj

ГОНТАРЕНКО Яна Дмитрівна
магістрантка спеціальності «Прикладна математика»
Черкаського національного університету імені Богдана Хмельницького
e-mail: yana.hontarenko@gmail.com

КРАСНОШЛИК Наталія Олександрівна
кандидат технічних наук, доцент,
доцент кафедри прикладної математики та інформатики
Черкаського національного університету імені Богдана Хмельницького
e-mail: wlik007@ukr.net
ORCID 0000-0003-4661-6997

ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ДІЙ ЛЮДИНИ ПО ВІДЕО

Розпізнавання дій людини по відео є важливою задачею в області комп'ютерного зору, яка знаходить широке застосування у різних сферах діяльності людини.

У даній роботі розглянуто методи розв'язання даної задачі з використанням штучних нейронних мереж. Описано два підходи застосування нейронні мереж: Transfer Learning та метод зміни простору. Метод Transfer Learning дозволяє використовувати досвід, отриманий під час розв'язання однієї задачі, для розв'язання іншої. Метод зміни простору полягає у використанні прогнозу попередньо-навченої моделі, як вхідних ознак для ще однієї нейронної мережі. У якості такої нейронної мережі обирали мережі різних архітектур з повнозв'язними шарами або шаром LSTM. Також були використані попередньо-треновані мережі MobileNet, ResNet та DenseNet.

Реалізацію розглянутих нейронних мереж здійснено за допомогою бібліотеки Keras. Для навчання моделей використано два типи вхідних даних: відео фрагменти та координати суглобів у просторі.

Для класифікації дій людини за координати суглобів у просторі також застосовували класичні алгоритми машинного навчання: метод найближчих сусідів, логістична регресія, випадковий ліс та метод опорних векторів.

Досліджено ефективність використання запропонованих моделей для розпізнавання дій людини по відео за долею правильних відповідей на тестовій виборці і часом навчання.

Ключові слова: *нейронна мережа, розпізнавання дій людини по відео, Transfer Learning, метод зміни простору, задача класифікації.*

Постановка проблеми

Розпізнавання дій людини на відео є важливою частиною систем штучного інтелекту. Незважаючи на великий прогрес у цьому напрямку, розпізнавання рухів все

ще залишається нерозв'язаною задачею у сфері комп'ютерного зору та є перспективною областю для подальших досліджень [1].

Однією з кінцевих цілей розвитку штучного інтелекту в цьому напрямку є створення машин, які б могли досконало розуміти людей, їхні емоції, стан, рухи, дії та наміри, для того, щоб краще аналізувати ситуації і виконувати відповідні запрограмовані дії.

Іншими важливими областями використання цієї задачі є, наприклад, відеоспостереження, сфера розваг та інші. Тут ключовими аспектами є обчислювальні алгоритми, які можуть розпізнавати людські рухи. Вони повинні позначати дію відповідним ярликом після спостереження цілої послідовності людських рухів, або її частини, аналогічно до людської системи зору. Побудову таких алгоритмів зазвичай досліджують у сфері комп'ютерного зору, яка вивчає як запрограмувати комп'ютери на розуміння зображень або послідовності зображень – відео.

Термін розпізнавання дій людини в сфері досліджень комп'ютерного зору коливається від простого руху кінцівки до комплексного руху декількох кінцівок та суглобів. Цей процес динамічний і як правило відображається у відео, що триває декілька секунд. З цих причин, важко описати задачу розпізнавання рухів людини в сфері комп'ютерного зору, проте є багато проектів вартих уваги [2].

Метою статті є використання нейронних мереж для розпізнавання дій людини по відео.

Методи розв'язання

1. Методи розв'язання задачі розпізнавання дій людини за допомого штучних нейронних мереж

Штучні нейронні мережі (ШНМ) – це математичні моделі, за основу в яких взято, біологічні нейронні мережі, а саме модель людського мозку. Мозок людини складається з нейронів, з'єднаних між собою синапсами. В свою чергу, штучна нейронна мережа складається з штучних нейронів - обчислювальних елементів. Аналогічно до людського мозку в штучних нейронних мережах є зв'язки, що визначають їх характеристики.

У нейронних мереж є такі переваги: паралелізм, можливість навчання та здатність до узагальнення. Оскільки кожен нейрон людського мозку одночасно зв'язаний із сотнями, тисячами інших нейронів, можна сказати, що він володіє високим ступенем паралелізму. Саме це є причиною того, що ШНМ підходять для вирішення задач, з якими раніше могла впоратись тільки людина [3].

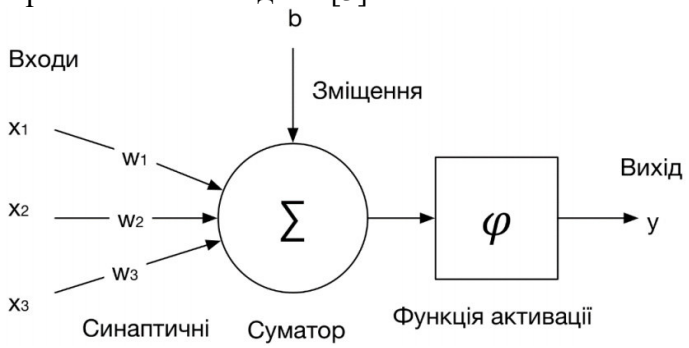

Рис. 1. Зображення математичної моделі ШНМ з одним прихованим шаром (суматор та функція активації)

Для реалізації штучних нейронних мереж використали бібліотеку Keras, яка є надбудовою бібліотеки TensorFlow. Вона в свою чергу використовує графічні

процесори від компанії Nvidia, в яких є вбудована архітектура CUDA – програмно-апаратна архітектура паралельних обчислень, яка дозволяє істотно збільшити обчислювальну продуктивність. Ці графічні процесори працюють набагато швидше звичайних центральних процесорів. На жаль, в мене немає можливості використання такого обладнання, тож було прийнято рішення використовувати Google Colaboratory.

Що б працювати з відеоданими, враховуючи обмеженість обчислювальних ресурсів, було прийнято рішення не працювати з відео, як цілісною структурою, а з кожним кадром цього відео, як з картинкою. Використовуючи цей метод, було обрано декілька підходів: transfer learning та метод зміни простору – використання попередньо тренованої моделі для прогнозу того, на що вона вчилась і потім використання цього результату як вхідних ознак для ще одної нейронної мережі.

Transfer Learning

Transfer Learning дозволяє використовувати досвід, що накопичувався під час вирішення одної задачі, для вирішення іншої. Часто цей метод використовується для класифікації зображень.

Бібліотека Keras надає доступ до багатьох уже реалізованих архітектур: ResNet, VGG та інших. Вони є попередньо-тренованими на базі даних ImageNet. ImageNet – проект по створенню великої бази даних розмічених зображень призначених для тестування методів розпізнавання образів та машинного зору. Із запропонованих архітектур було обрано MobileNet, ResNet та DenseNet.

MobileNet (рис. 2) вважається однією з кращих нейронних мереж. Вона має небагато параметрів, швидка та може прогнозувати з точністю, яку можна прирівняти до моделей такого рівня як ResNet або VGG. Це дає можливість використовувати мінімум ресурсів для гарного результату.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Рис. 2 Архітектури нейронної мережі MobileNetV1 [4]

MobileNetV1 складається з одного звичайного згорткового шару 3x3, який використовує шар «Batch Normalization» та функцію активації «Relu», та тринадцяти блоків з шести елементів: поглиблений згортковий шар 3x3 з шаром «Batch Normalization» та «Relu» та звичайний згортовий шар 1x1, шар «Batch Normalization» та «Relu». Особливістю її архітектури є те, що в ній немає pooling-шарів. Саме цю нейронну мережу буде використано для експериментів.

ResNet (residual neural network) – залишкова нейронна мережа. Це означає, що вона має глибоку залишкову структуру навчання (рис. 3), додаються з'єднання швидкого доступу. Вони пропускають один або декілька шарів і виконуються співставлення ідентифікаторів цього прошарку та попереднього залишкового.

Архітектура ResNet50 має наступну структуру: згортковий шар 7x7, шар max-pool, блоки згорткових шарів з розміром 1x1, 3x3, 1x1, кожен з який відрізняється від попереднього їх кількістю в мережі та кількістю нейронів у кожному згортковому шарі.

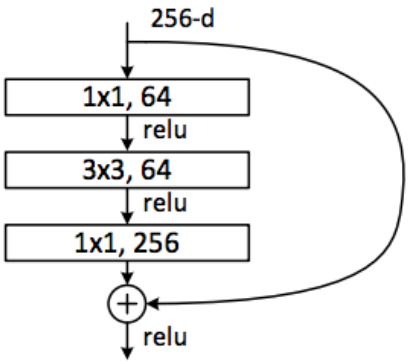

Рис. 3 Перший блок згорткових шарів в ResNet50 [5]

DenseNet – це нейронна мережа, яка не має дуже значних досягнень в сфері глибокого навчання, але її архітектура є досить цікавою (рис. 4). Як видно, кожен з виходів блоку з'єднаний з входом в наступні. Така архітектура дозволяє зменшити ризик того, що не зійдеться градієнт, та зменшує кількість параметрів тощо.

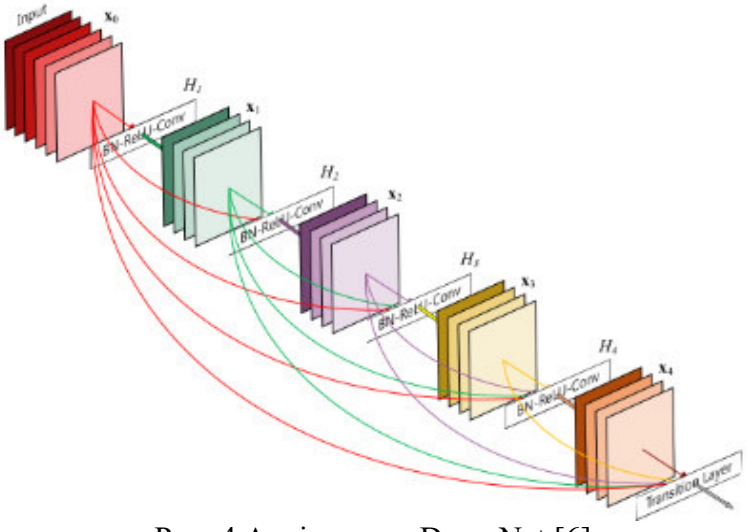

Рис. 4 Архітектура DenseNet [6]

DenseNet121 має послідовність з 7x7 згорткового шару, шару max-pool та Dense блоку, 1x1 згорткового шару, average-pool, що повторюються 3 рази. Після цього йде

ще один Dense блок. Цей блок в собі містить два згорткових шари розмірами 1x1 та 3x3 відповідно. Але в кожному блоці різна кількість такого набору шарів.

Метод зміни простору

Метод зміни простору передбачає використання прогнозу попередньо-навченої моделі, як вхідних ознак для ще одної нейронної мережі.

У цьому випадку було побудовано декілька варіантів нейронної мережі, які поділялись на два типи: нейронні мережі з повнозв'язними шарами та шаром LSTM. Для кожного з цих типів було побудовано декілька варіантів. Для моделі з повнозв'язними шарами архітектура будувалась таким чином:

- визначався шар, який перетворює тензор вхідних ознак;
- обиралась кількість повнозв'язних шарів;
- підбирались значення в шарі dropout.

Для моделі з шарами LSTM підбирались тільки значення в шарі dropout.

2. Методи розв'язання задачі розпізнавання дій людини за допомого класичних алгоритмів класифікації

Ще одним підходом до розпізнавання дій людини по відео є застосування алгоритмів класифікації даних. Ці алгоритми можна застосувати для класифікації рухів за розміщенням суглобів у просторі.

Задача класифікації – це задача розбиття множини об'єктів на задані групи, всередині кожної з яких вони вважаються схожими один на одного. Основні проблеми, з якими зустрічаються при розв'язку задач класифікації, – це незадовільна якість вхідних даних. Також алгоритми стандартного машинного навчання, які працюють з задачею класифікації дуже чутливі і легко піддаються проблемам перенавчання і недонавчання. В першому випадку модель дуже підлаштовується під всі дані, на яких тренується, під всі аномалії які там є, і не зможе правильно описати тестові дані, тому що не матиме узагальнюючої здатності. Терміном недонавчання позначають ситуацію, коли модель, що використовувалась, не змогла навчитись на тренувальній вибірці, виділити важливі ознаки, класифікувати її об'єкти [7].

Було використано наступні алгоритми класифікації.

Метод k-найближчих сусідів – це один з найпростіших методів класичного машинного навчання. Він працює за наступним алгоритмом для кожного об'єкту з набору даних:

1) обчислити відстань до всіх інших об'єктів;
2) обрати k елементів відстань до яких мінімальна. Це й будуть сусіди об'єкта;
3) класом об'єкту буде той клас, який найчастіше зустрічається в сусідів [8].

Логістична регресія дозволяє оцінити ймовірність належності об'єкту до певного класу. Основна задача логістичної регресії, побудувати пряму (гіперплощину) так, щоб вона максимально розділяла класи, що знаходяться на цій площині (в цьому просторі).

Випадковий ліс – це ансамблевий метод навчання, за основу якого взято побудову визначеної кількості дерев рішень.

Нехай у нас є вибірка з N елементів, M ознак та параметр m. Для всіх дерев рішень в ансамблі будуть, незалежно один від одного, виконуватись такі дії:

1) генерується підвибірка з повторенням розміром n з навчальної вибірки;
2) будується дерево рішень, яке класифікує елементи даної підвибірки, за m ознаками;
3) дерево будується до повного вичерпання підвибірки та не піддається процедурі відсікання.

Після цього проводиться класифікація об'єктів: кожне дерево відносить елемент, для якого проводилась класифікація до одного з класів та визначається той клас, за який проголосувало найбільше дерев [9].

Метод опорних векторів – метод класифікації, що використовує розділяючу пряму або гіперплощину. Він відрізняється від інших гіперплощинних методів класифікації (таких, як логістична регресія) тим, що дозволяє обирати оптимальне розташування гіперплощини. Вона обирається таким чином, щоб бути на максимальній відстані від елементів кожного з класів.

Результати дослідження

1. Пошук та підготовка даних

Перша задача, яку потрібно було вирішити – це пошук даних, які будуть використовуватись для розпізнавання дій людини по відео. Необхідно було знайти два набори даних. Перший набір мав містити відео з різними діями людей. Другий набір повинен був бути розмічений, тобто містити координати, за якими суглоби розміщувались у просторі.

Для першого випадку було обрано набір даних Kinetics-600 (рис. 5). Він містить 500 000 відео та 600 рухів. Це один з найкращих та найбільших наборів даних для задач такого типу.

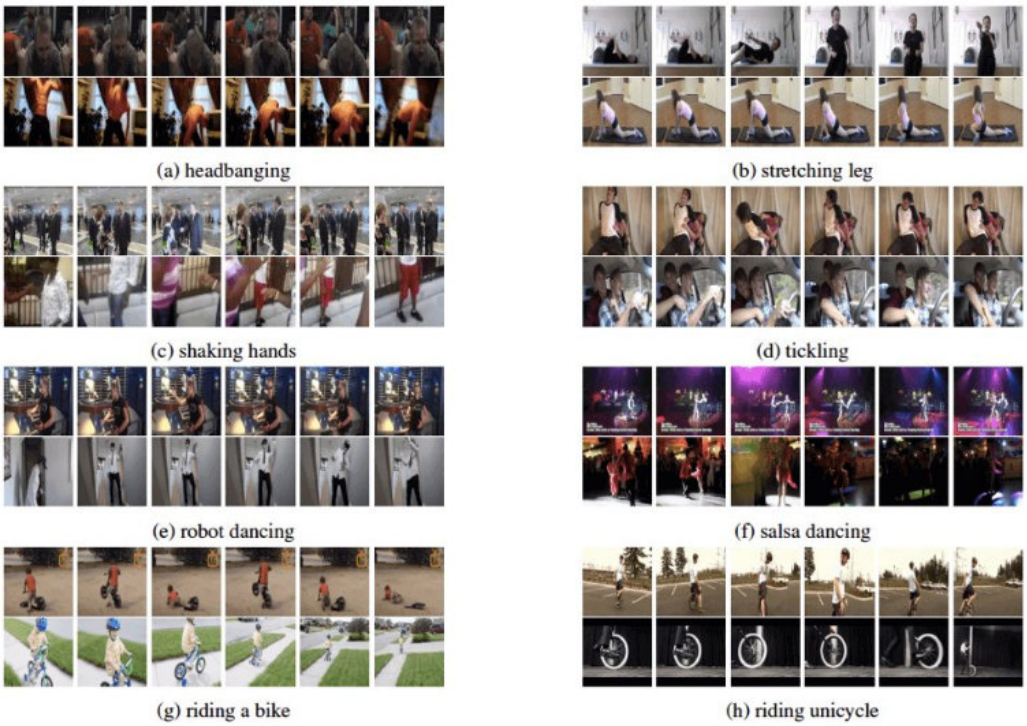

Рис. 5 Ілюстрація деяких відео з набору даних Kinect-600

Даний набір даних для обробки потребує великого об'єму пам'яті та значних обчислювальних ресурсів. Для подальших досліджень було зменшено кількість використаних відео. Випадковим чином обрали 50 дій із 600, і на кожну дію було виділено певну кількість відео. В результаті було збережено близько 22800 відео для тренування, 2500 відео для валідації та 4650 відео для тестування.

Для другого набору даних ключовим було питання про те, як його використовувати. Такі дані отримуються завдяки спеціальним RGB-D камерам або сенсорам. Вони мають спеціальний сенсор глибини, який може визначити, наскільки

далеко об'єкт знаходиться від нього. Завдяки цьому, вони можуть відслідковувати людину, її рухи, її образ із постійного потоку відео. Таким чином, можна це інтерпретувати так, із потоку RGB відео виділяються кадри. З потоку «глибинного» розділеного на кадри відео прибирається шум. Потім ці кадри об'єднуються у картинки. Далі відбувається процес витягування ознак з цих даних та сама класифікація.

Для застосування алгоритму класифікації рухів за допомогою RGB-D послідовності було обрано набір даних UTKinect-Action3D. Він містить дані 10 людей, які виконують 10 різних дій по два рази. Все це зберігається в трьох архівах:

– набір RGB-картинок;

– набір «глибинних» картинок;

– набір координат 20 точок (рис. 6) суглобів в просторі.

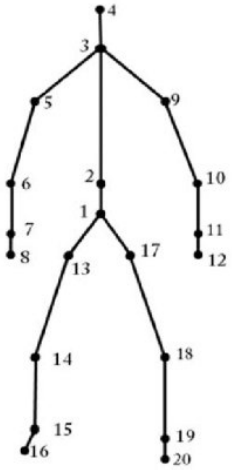

Рис. 6 Нумерація суглобів на тілі людини

Вважається, що набір координат це і є ознаки, які потрібно виявити, тож з даної задачі буде вирішуватись тільки підзадача класифікації.

2. Попередня обробка даних

Попередня обробка даних передбачала попередню обробку відео фрагментів та попередню обробку координат суглобів.

При попередній обробці відео фрагментів необхідно було представити відео у вигляді тензору – багатовимірної матриці. Для цього потрібно було зробити такі кроки:

1) завантажити відео з сервісу YouTube;

У наборі даних, що було використано була така інформація: посилання на розміщення відео на ресурсі YouTube, секунда ролика на якому починалась дія, секунда на якій вона закінчувалась, та назва цієї дії.

2) обрізати розмір відео (до 10 відведених секунд) та зберегти цей відрізок відео у новому розширенні з іншим FPS;

Змінювати розширення та кількість кадрів в секунду потрібно було для того, щоб потім отримувати матриці однакового розміру.

3) завантажити отримане нове відео, нормалізувати його і зберегти у вигляді матриці, для подальшого використання.

Отримані файли містять тензори розміром (50, 20, 200, 200, 3). Тобто це – (кількість відео, секунди * кількість кадрів, ширина, висота, канали кольору).

Метою попередньої обробки координат суглобів також було отримання матриць. З самого початку одне відео містить всі 10 дій. Назва цих дій та їх час (номера кадрів з

ними) вже містяться у наборі даних. Головним завданням було нормалізувати довжину дії, та, відповідно, і розмір майбутньої матриці. Наявні у наборі дії займають різну кількість кадрів. Тобто є дії, які відмічені менш ніж на 10 кадрах, а є ті, що займають більше 100. Для стабілізації було обрано 80 кадрів. Якщо в послідовності кадрів, що зображає дія їх не вистачало, то вони дублювались. Якщо ж їх було забагато, то послідовність обрізалась з двох кінців. Після цього відповідно до кожного елементу нової послідовності будувалась матриці з координат суглобів у просторі, які теж були попередньо отримані з текстового файлу у змінну. У результаті на кожне відео одного користувача було збережено матрицю розміром (10, 80, 20, 3), де 10 – кількість дій, 80 – послідовність кадрів, 20 – кількість суглобів та три їх координати.

3. Реалізації методів роботи з послідовністю картинок

Transfer Learning

Для методу transfer learning було використано моделі MobileNetV1, DenseNet121 і ResNet50. Важливою частиною є завантаження даних в саму модель. Враховуючи обмеженість обчислювальних ресурсів використовували генератори для формування даних. Головна ідея використання генераторів у тому, щоб не завантажувати весь набір даних за раз. Можна завантажити його частинами за певну кількість кроків. Отримати частину тренувальних даних, частину відповідей на них та віддавати це на вхід моделі.

Процес розпізнавання дій по відео передбачає роботу з картинками і класифікацію рухів, опираючись на прогноз для всіх картинок, які цей рух зображали. Тобто, для того, щоб спрогнозувати дію на одному відео потрібно було зробити прогноз для 20 картинок. В процесі «розкладання» відео на картинки кожна матриця змінює форму з матриці форми (кількість відео, кадри в секунду * секунди, ширина, висота, канали кольорів) на матрицю форми (кількість відео * кадри в секунду * секунди, ширина, висота, відео).

Також деякі перетворення проводяться і для відповідей. Кожна відповідь замінюється на вектор з нулів, в якому одиничка стоїть на тому місці, номер якої має відповідь (3 це [0, 0, 0, 1, 0, 0]). Оскільки для використаного методу нам потрібно мати одну відповідь на 20 елементів, то він дублюється 20 разів. Всі ці перетворення виконувались для кожного файлу завдяки генератору і не займали всю пам’ять.

Було використано три попередньо-навчені моделі. Фрагмент коду для роботи з цими моделями наступний:

```
def model(input_shape):
    inp = Input(input_shape)
    base_model = applications.MobileNet(include_top=False, weights='imagenet', input_shape=input_shape,
pooling='avg')
    base_model.trainable = False
    resnet = base_model(inp)
    drop = Dropout(0.7)(resnet)
    fc = Dense(50)(drop)
    out = Activation('softmax')(fc)
    return Model(input=inp, output=out)
model = model((WIDTH, HEIGTH, 3))
model.compile(optimizer='adam',  loss='categorical_crossentropy', metrics=['accuracy'])
```

Спочатку створюється функція, яка буде повертати об’єкт типу Model. Це спеціальний об’єкт бібліотеки Keras, який дозволяє створити об’єкт нейронної мережі з усіма зв’язками, які пов’язують вхідний та вихідний шар. У самій функції записуються шари нейронної мережі від вхідного шару до вихідного. У даному випадку ми отримуємо на вхід об’єкт з певним розміром. Запам’ятовуємо цей об’єкт у першому вхідному шарі. Далі завантажуємо попередньо-треновану модель: параметр include_top вказує на те, чи ми будемо використовувати її класифікатор (include_top=True) чи

будемо додавати свій (include_top=False), weights – які ваги будемо використовувати та задається розмір даних, які подаються на вхід. Після цього вказуємо, що тренувати модель не будемо, додаємо свій класифікатор, тобто вказуємо нашу кількість класів у вихідному шарі та функцію активації, яку хочемо використовувати.

Після чого ми створюємо об'єкт описаної моделі – це робиться викликом функції, та вона компілюється. Параметри компіляції були обрані з рекомендацій, найчастіше використовують саме такі параметри для задач мульти-класифікації. Останнім етапом є тренування.

Метод зміни простору

За допомогою створеного раніше генератору та попередньо-тренованих моделей, до яких надає доступ бібліотека Keras було зроблено прогнози, але не конкретних дій, а того, що вміє дана модель (MobileNetV1) прогнозувати. Ідея полягає в тому, що нейронна мережа змінює простір і вже на іншому просторі буде навчатись для розпізнавання дій. Можна інтерпретувати це так: перша нейронна мережа розпізнає певні предмети на відео, а за допомогою її прогнозів інша вчиться відповідати на питання що за дія на відео.

Для цього спочатку завантажується файл, який містить декілька відео. Для кожного з цих відео робиться прогноз на завантаженій моделі та зберігається в окремому файлі. Таким чином, для одного відео буде вже не матриця розміром (кількість кадрів в секунду * секунди, ширина, висота, канал кольору), а (кількість кадрів в секунду * секунди, 1000). Також паралельно з цим знову зберігались і позначки на ці відео, але вже по одному елементу. Після того, як уже було створено такі вхідні дані, вони були завантажені у матриці і за допомогою генератора, який поділяв ці матриці на частини, подавались на вхід моделям двох типів: що вміщали в собі тільки повнозв'язні шари та модель з шаром LSTM.

4. Реалізація методів роботи з координатами суглобів у просторі

Нейронні мережі

Для використання штучних нейронних мереж при класифікації рухів за розміщенням суглобів у просторі, потрібно було представити дані у вигляді картинки за наступним алгоритмом:

- кожну координату було перетворено за формулою

$$(x_i', y_i', z_i') = F(x', y', z') \ ,$$

$$x_i' = 255 \cdot \frac{x_i - \min\{C\}}{\max\{C\} - \min\{C\}}, \ y_i' = 255 \cdot \frac{y_i - \min\{C\}}{\max\{C\} - \min\{C\}}, \ z_i = 255 \cdot \frac{z_i - \min\{C\}}{\max\{C\} - \min\{C\}},$$

де $\min\{C\}$ та $\max\{C\}$ – мінімум та максимум серед усіх координат набору даних. Після перетворення дані зберігаються у матрицях розміру (80, 20, 3), але в форматі RGB;

- змінюємо чергування груп суглобів – нова послідовність $P_1 \to P_2 \to P_3 \to P_4 \to P_5$.

В результаті цієї операції змінюється розмір матриці, тоді отримуємо матрицю (40, 40, 3) [1].

Класичні методи машинного навчання

Для використання класичних методів машинного навчання при класифікації рухів за розміщенням суглобів у просторі, потрібно було представити дані у вигляді матриці, рядками якої були б самі дії. Тобто в результаті обробки даних потрібно отримати матрицю розміру (200, 80 * 20 * 3).

Для всіх зазначених раніше методів машинного навчання виконували наступні кроки:

- поділити матрицю на два фрагменти: для тренування та для тестування;
- застосувати класичні методи машинного навчання для тренування;
- перевірити результати;
- підібрати параметри для використаних методів;
- перевірити результати для кращих параметрів.

5. Результати проведених обчислювальних експериментів

Результати навчання попередньо-тренованих моделей продемонстрували, що мережа ResNet відразу почала перенавчатись, DenseNet з самого початку дала непоганий результат, але найкраще спрацювала мережа MobileNet. Тому її обрали для методу зміни простору, щоб отримати прогнози, які будуть вхідними ознаками для наступної нейронної мережі. У якості цієї мережі розглянули повнозв'язні нейронні мережі та нейронні мережі з шаром LSTM різної архітектури. Процес навчання найкращих з них представлено на рис. 7., з якого видно, що хоч модель з шаром LSTM навчалась значно довше, але все одно продемонструвала гірші результати, ніж модель з повнозв'язними шарами, хоча і ця модель почала перенавчатись і не дала бажаного результату.

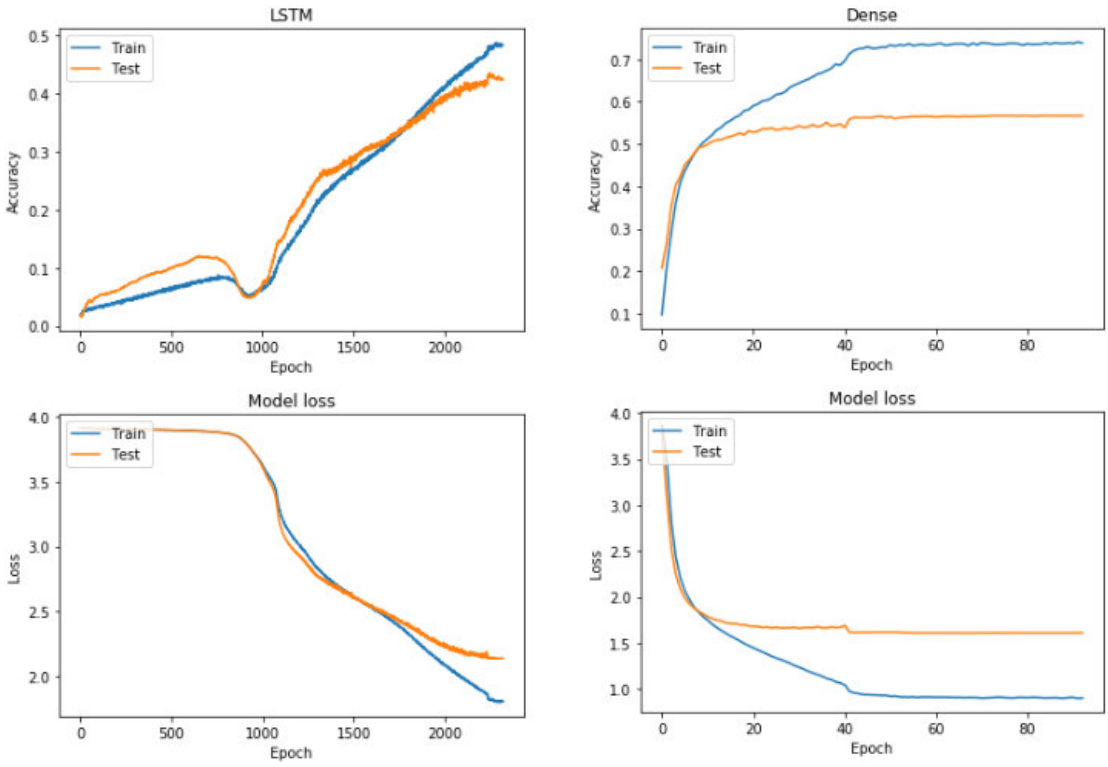

Рис. 7 Процес навчання моделей у зміненому просторі

Таблиця 1

Метод та модель	Час навчання для однієї епохи	Кількість епох	Доля правильних відповідей на тестовій виборці
Попередньо-тренована модель ResNet50	540 с	2	0.018750
Попередньо-тренована модель DenseNet121	520 с	14	0.505625

Продовження таблиці 1

Попередньо-тренована модель MobileNetV1	430 с	7	0.555625
Зміна простору, повнозв'язна модель	1 с	63	0.546076
Зміна простору, модель з шаром LSTM	9 с	2271	0.422025

Результати проведених обчислювальних експериментів наведено у табл. 1.

Далі розглянемо метод роботи з координатами суглобів у просторі. Було проведено експерименти з різними типами нейронних мереж, але в результаті було обрано згорткову нейронну мережу з декількома блоками. Процес навчання нейронної мережі для цих даних представлено на рис. 8.

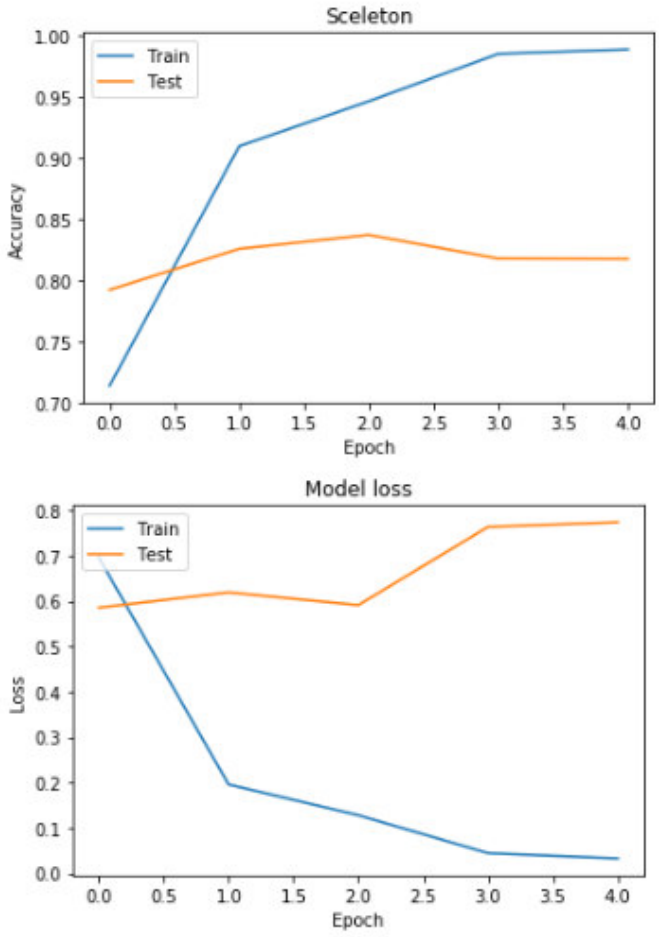

Рис. 8 Процес навчання згорткової нейронної мережі для форматованих координат суглобів

З наведеного графіку видно, що дані добре оптимізуються і з перших кроків мережа добре навчається.

Також для цього типу даних було використано і класичні методи машинного навчання. Спочатку було проведено обчислювальні експерименти для загальних моделей, а потім підібрано параметри для кожної них. Остаточні результати представлено у табл. 2.

Таблиця 2

Метод та модель	Час навчання	Доля правильних відповідей на тестовій виборці
Згорткова нейронна мережа	50 с	0.80
Метод найближчих сусідів	7.15 мс	0.55
Логістична регресія	7.15 мс	0.85
Випадковий ліс	5.25 мс	0.58
Метод опорних векторів	6.68 мс	0.80

В результаті проведеного дослідження дійшли висновку, що реалізувати розпізнавання дій людини без значних обчислювальних ресурсів досить важко. Також потрібно багато пам'яті, щоб була можливість завантажувати всі дані в ході виконання програми, а не порціями, як це було реалізовано.

Висновки

У роботі розглянуто методи розв'язання задачі розпізнавання дій людини по відео. Для цього було використано два типи даних: відео та координати суглобів у просторі.

Для кожного типу даних було використано декілька типів нейронних мереж.

Для набору даних з відео було реалізовано два підходи: у першому було використано три попередньо-треновані моделі; у другому, після зміни простору було використано 2 типи моделей. Найкращі результати за долею правильних відповідей на тестовій виборці було отримано для попередньо-тренованої моделі MobileNetV1. Також хороші результати отримано за допомогою повнозв'язної нейронної мережі, використовуючи метод зміни простору.

Для набору даних з координатами суглобів у просторі проводили експерименти з різними архітектурами нейронної мережі: з повнозв'язними шарами та згортковими. Також для цих даних використали класичні методи машинного навчання: метод найближчих сусідів, логістична регресія, випадковий ліс та метод опорних векторів. Серед класичних модей найкращі результати на тестовій виборці було отримано для логістичної регресії.

Отже, розглянуті моделі мали непоганий результат враховуючи, що у розглянутій вибірці було 50 різних типів рухів.

Також варто зазначити, що для реалізації моделей розпізнавання дій людини, потрібна значна попередня обробка даних. Мається на увазі, що багато часу займає завантаження та обробка відео і вибір моделей для роботи з ними.

Враховуючи те, що кращі результати отримали для моделей які працювали з координатами суглобів у просторі, можна сказати, що було б доцільно реалізувати модель, яка працюватиме з потоком відео та автоматично виділятиме точки суглобів на тілі людини і аналізуючи їх могла визначити, що за дію виконує людина.

Список використаної літератури:

1. Pham, H.-H. Learning and Recognizing Human Action from Skeleton Movement with Deep Residual Neural Networks / H.-H. Pham, L. Khoudour, A. Crouzil, P. Zegers, S.A. Velastin. [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/1803.07780.pdf
2. Yu, K. Human Action Recognition and Prediction: A Survey / K. Yu, F. Yun // Journal of latex class files. – 2018. – Vol. 13, No. 9. – 1-20 p.

3. Мошенченко, М. С. Штучні нейронні мережі / М. С. Мошенченко // Topical problems of modern science. – 2017. – Vol.4. – С. 47-49.
4. Akogo, D.A. CellLineNet: End-to-End Learning and Transfer Learning For Multiclass Epithelial Breast cell Line Classification via a Convolutional Neural Network / D.A. Akogo, V. Appiah, X.-L. Palmer. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/327134257_CellLineNet_End-to-End_Learning_and_Transfer_Learning_For_Multiclass_Epithelial_Breast_cell_Line_Classification_via_a_Convolutional_Neural_Network
5. He, K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1512.03385
6. Сайт PyTorch [Електронний ресурс]. – Режим доступу: https://pytorch.org/hub/pytorch_vision_densenet
7. Біла, Н.І. Інформаційні системи та технології в управлінні: методичні вказівки / Н.І. Біла. – Запоріжжя: ЗНТУ, 2014. – 50 с.
8. Классификатор kNN / Хабр [Електронний ресурс]. – Режим доступу: https://habr.com/ru/post/149693
9. Hastie T. The Elements of Statistical Learning / T. Hastie, R. Tibshirani, J. Friedman // Chapter 15. Random Forests. – 2009. – 587-623 p.

Bibliography:

1. Pham, H.-H., Khoudour, L., Crouzil, A., Zegers, P., & Velastin, S.A. (2018). Learning and Recognizing Human Action from Skeleton Movement with Deep Residual Neural Networks. Retrieved from https://arxiv.org/pdf/1803.07780.pdf
2. Yu, K., Yun, F. (2018). Human Action Recognition and Prediction: A Survey. *Journal of latex class files*, 13(9), 1-20.
3. Moshenchenko, M. S. (2017). Shtuchni neironni merezhi [Artificial Neural Nets]. *Topical problems of modern science*, 4, 47-49 [in Ukrainian].
4. Akogo, D.A., Appiah, V., & Palmer, X.-L. (2018). CellLineNet: End-to-End Learning and Transfer Learning For Multiclass Epithelial Breast cell Line Classification via a Convolutional Neural Network. Retrieved from https://www.researchgate.net/publication/327134257_CellLineNet_End-to-End_Learning_and_Transfer_Learning_For_Multiclass_Epithelial_Breast_cell_Line_Classification_via_a_Convolutional_Neural_Network
5. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. Retrieved from https://arxiv.org/abs/1512.03385
6. PyTorch *pytorch.org*. Retrieved from https://pytorch.org/hub/pytorch_vision_densenet
7. Bila, N.I. (2014), Informatsiini systemy ta tekhnolohii v upravlinni: metodychni vkazivky [Information Systems and Technologies in Management: Guidelines]. Zaporizhzhia: ZNTU [in Ukrainian].
8. (2012). Klassifikator kNN *habr.com*. Retrieved from https://habr.com/ru/post/149693
9. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Chapter 15. Random Forests, 587-623.

HONTARENKO Yana,
Student, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine
KRASNOSHLYK Nataliia,
Candidate of Technical Sciences, Associate Professor, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

HUMAN ACTION RECOGNITION FROM VIDEOS USING NEURAL NETWORKS

Summary. Introduction. *Human action recognition is an important problem in the field of computer vision, which is widely used in various areas of human activity.*

This paper describes methods for solving this problem using neural networks. Two approaches to using neural networks are described: Transfer Learning and Space Modification. The Transfer Learning method allows you to use the experience gained when solving one task to solve another. The method of changing space is to use the prediction model learned as inputs for another neural network. As such a neural network was chosen networks of different architectures with fully connected layers or LSTM layer. MobileNet, ResNet and DenseNet pre-trained networks were also used. Two types of input are used to train models: video snippets and joint coordinates in space.

Classical machine learning algorithms were also used to classify human actions by the coordinates of joints in space: the nearest neighbor method, logistic regression, random forest, and support vector machine.

The Purpose *of this paper is to use the neural networks to human action recognition from videos.*

Results. *The neural networks considered were implemented using the Keras library.*

Pre-processing of data is performed to represent them in the form of multidimensional matrices (tensors). Several types of neural networks were used for each data type.

The effectiveness of the use of the proposed models for the recognition of human actions on video by the share of the correct answers on the test sample and the time of training is investigated.

The best results in the proportion of correct answers were obtained for the MobileNetV1 pre-trained model. Also, good results were obtained using a fully connected neural network using the space-changing method. Among the classical machine learning algorithms, the best results on the test sample were obtained for logistic regression.

The models presented show a good result considering that there were 50 different types of human movements in the sample used.

Conclusion. *This paper describes the problem of human action recognition. Neural networks were used to solve this problem. Various approaches have been implemented to use neural networks to recognize human actions through video, and computational experiments have been conducted. Classical machine learning algorithms have been applied to classify joint motion in space.*

The study found that human action recognition from videos requires significant computing resources. Significant pre-processing is also required.

Considering that the best results were obtained for models that worked with joint coordinates in space, it can be concluded that it would be advisable to implement a model that will work with the video stream and automatically highlight the joint points on the body.

Keywords: *neural network, human action recognition from videos, transfer learning, space change method, classification.*

Одержано редакцією 19.07.2019 р.
Прийнято до публікації 09.10.2019 р.

УДК 538.9

DOI 10.31651/2076-5886-2019-2-72-86

PACS 07.05.Tp, 61.43.Bn, 64.60.De, 66.30.-h, 82.60.Lf

ПАСІЧНИЙ Микола Олександрович
кандидат фізико-математичних наук,
доцент, завідувач кафедри фізики,
Черкаський національний університет
імені Богдана Хмельницького
e-mail: pasichnyy@ukr.net
ORCID: 0000-0002-8434-1544

ДОСЛІДЖЕННЯ КРИВИХ ФАЗОВОЇ РІВНОВАГИ БІНАРНИХ ГЦК-СПЛАВІВ З ОБМЕЖЕНОЮ РОЗЧИННІСТЮ КОМПОНЕНТІВ РЕШІТКОВИМИ МЕТОДАМИ МОНТЕ-КАРЛО

У роботі проведене дослідження кривих фазової рівноваги бінарних ГЦК-сплавів з обмеженою розчинністю компонентів на основі решіткових методів Монте-Карло. Підтверджено існування єдиної фазової діаграми модельної системи, що не залежить від конкретного дифузійного механізму та ймовірнісного алгоритму. Виявлено лінійні залежності різниці приведених температур кривих фазової рівноваги від концентрації для моделі регулярного твердого розчину та модельної бінарної системи на основі решіткових методів Монте-Карло.

Ключові слова: *Монте-Карло, бінарний сплав, регулярний твердий розчин, метод дифузійної пари.*

Вступ

Поява і стрибкоподібний розвиток ЕОМ став стимулом для удосконалення широкого класу чисельних методів для моделювання фізичних процесів. Обсяги оперативної пам’яті та швидкодія сучасної обчислювальних систем дають змогу