УДК 004.93

А.М. Літвинчук, Л.В. Барановська

ПОКРАЩЕННЯ МОДЕЛЕЙ РОЗПІЗНАВАННЯ ОБЛИЧ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ, НАВЧАННЯ ПОДІБНОСТІ ТА МЕТОДІВ ОПТИМІЗАЦІЇ

Ключові слова: згорткові нейронні мережі, розпізнавання облич, навчання подібності, методи оптимізації.

Ключевые слова: сверточные нейронные сети, распознавание лиц, обучение подобия, методы оптимизации.

Вступ

Розпізнавання облич — це одна з основних задач комп'ютерного зору. Вона має безліч прикладних застосувань, що призвело до величезної кількості досліджень у цій сфері. І хоча дослідження проводились з початку розвитку комп'ютерного зору, адекватних результатів змогли досягнути лише за допомогою згорткових нейронних мереж.

Нейронні мережі виникли в 60-х роках XX століття, і з того часу сфера їх дослідження мала як злети, так і падіння. Проте з розвитком обчислювальних потужностей, зокрема графічних карт, відбулось широке розповсюдження нейронних мереж. Графічні карти, на відміну від простих процесорів, мають тисячі ядер, потужність яких набагато менша, ніж у процесорних, проте вони дозволяють розпаралелювати в тисячі раз більше операцій. Так, починаючи з 2010 р., люди можуть навчати нейронні мережі з багатомільйонною кількістю параметрів, а нині — навчати нейронні мережі можна на домашньому комп'ютері. Зокрема, широкого резонансу у розпізнаванні образів здобула концепція згорткової нейронної мережі, за допомогою якої у 2014 р. здобули перемогу у конкурсі розпізнавання образів ILSVRC (ImageNet Large-Scale Visual Recognition Challenge), показавши результати, кращі за будь-які попередні методи комп'ютерного зору.

Метою роботи є дослідження різних методів розпізнавання облич за допомогою згорткових нейронних мереж, різних способів оптимізації параметрів мереж, збільшення узагальнюючих можливостей моделей, а також їх комбінування задля визначення якісного і надійного алгоритму навчання нейронних мереж для задачі розпізнавання облич.

Сфера розпізнавання облич до нейронних мереж

Розглянемо сферу розпізнавання облич до нейронних мереж, а також покажемо мінуси кожного з підходів.

1. Метод головних компонент — це один з найвідоміших та найбільш якісно опрацьованих методів розпізнавання. Він широко відомий у стандартному машинному навчанні, зокрема використовується для пониження розмірності вхідних векторів ознак для більш ефективного та точного вирішення задачі.

Зменшення вектора розмірності обумовлено нашим наміром використати обличчя, отримати вектор базових ознак, які можуть бути спільними у різних людей. Цей підхід є основою для багатьох методів, зокрема також для згорткових нейронних мереж, проте кожний метод має свій спосіб отримання вектора базових ознак. Метод головних компонент аналізує набір тренувальних даних та виявляє зміну вхідних векторів ознак, після чого описує цю зміну в базисі власних векторів. Таким чином, власне число при власному векторі буде показувати його важливість.

Хоча метод математично добре обгрунтований та досліджений, на жаль, його результати у реальних задачах досить погані. Його плюсом є лише швидкість роботи — множення вектора на матрицю — це дуже швидка операція, в результаті метод головних компонент може обробляти тисячі фотографій в секунду. Проте він недостатньо місткий для хорошого розпізнавання великої кількості облич, не зважаючи на те, що покращується з великою кількістю вхідних даних. Також метод нестійкий до різних деформацій обличчя та шуму, часто присутньому у фотографіях.

2. Активні моделі зовнішнього вигляду, як і метод головних компонент, є статистичними моделями, які за рахунок деформацій підганяються під реальне зображення.

Ця модель включає в себе два типи параметрів: параметри, пов'язані з формою, а також параметри, пов'язані зі статистичною моделлю пікселів зображення та текстурою. Щоб навчити цю модель, потрібна повністю ручна розмітка даних. Кожне обличчя ми розбиваємо приблизно на 70 характерних точок, які модель буде вчитись адаптувати до нового зображення.

За допомогою активних моделей зовнішнього вигляду можна моделювати фотографії об'єктів з різними деформаціями. Частина параметрів моделює форму лиця, а частина — текстуру. Деформація у цьому випадку — масштабування, перенесення, поворот [1]. Цей метод можна використовувати як для детектування обличчя, так і для розпізнавання.

З явних плюсів можна виділити швидкість роботи, оскільки модель має не дуже багато параметрів. Також модель більш стійка до трансформацій, ніж метод головних компонет. Проте вона, як і метод головних компонент, не є місткою, а значить, і точною, не зважаючи на те, що модель статистична.

Опис підходів

Розглянемо архітектури нейронних мереж, підхід для навчання подібності та методи оптимізації, з якими ми експерементували.

1. Архітектури нейронних мереж. У даній роботі використано три архітектури нейронних мереж, які вважаються найуспішнішими відносно точності до швидкості: SE-ResNet50, EfficientNet-B0 та EfficientNet-B3.

Спочатку розглянемо ResNet50, її схема показана на рис. 1 [2, 3].

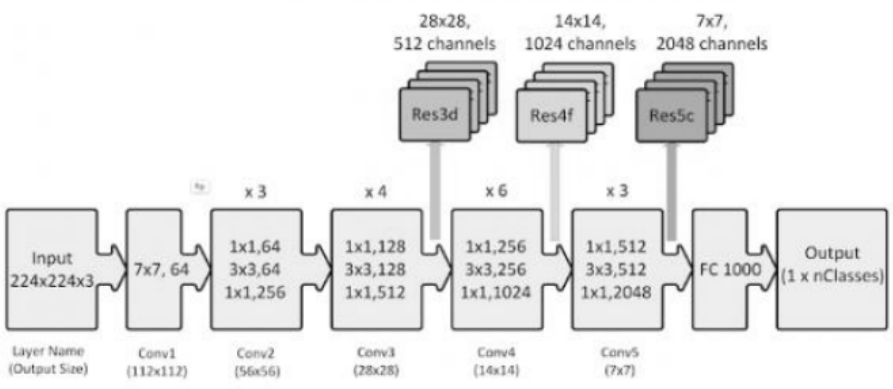

Рис. 1

Архітектура вводить до стандартної згорткової мережі так звані пропускні зв'язки. Ідея полягає в тому, щоб вихід певного набору шарів з'єднати з його входом. Розглянемо наступий приклад: нехай f — набір певних шарів, а $f(x)$ — перетворення, що роблять ці шари з вхідними даними. Тоді вихід шару разом з пропускним з'єднанням буде таким:

$$Output(x) = f(x) + x.$$

Завдяки цій операції можна гарантувати, що затухання градієнтів через шар $f(x)$ ніколи не відбудеться.

Архітектура SE-ResNet50 вносить до ResNet50 блок, який ми називатимемо блоком уваги. Його основна ідея — дати мережі можливість контролювати, які з каналів залишити після виконання згортки. Умовну схему блоку уваги [4] зображено на рис. 2.

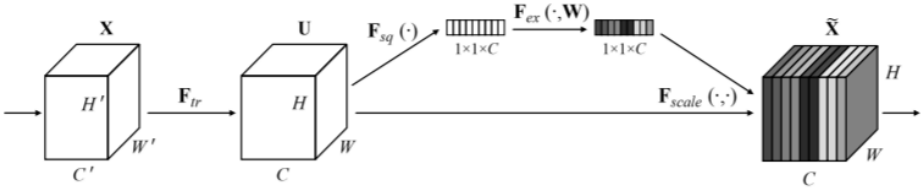

Рис. 2

Завдяки цій операції кожен канал множиться на число в проміжку [0, 1] (це число залежить від різних факторів — кореляції ознак у каналах тощо) і мережа зможе сама обрати його важливість. Цей блок став дуже популярним завдяки своїй простоті та відносно невеликому зменшенню швидкості роботи, при цьому суттєво збільшуючи точність класифікації.

Далі розглянемо EfficientNet-B0 та EfficientNet-B3. Вони належать до однієї сім'ї архітектур, а відрізняються лише глибиною (кількість шарів) та шириною шарів (кількість каналів у кожному шарі), тому розглянемо лише EfficientNet-B0 [5, 6]. Його умовна схема зображена на рис. 3.

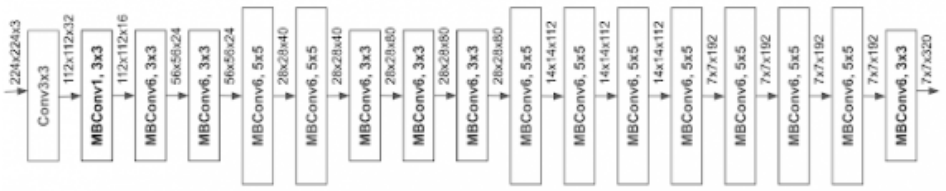

Рис. 3

Ця архітектура — більш ефективна версія ResNet, її ідея полягає у максимальному зменшенні навчальний параметрів та кількості обчислень. Це досягається за рахунок більш ефективної реалізації згортки, а також при оперуванні на зображеннях меншою кількістю каналів.

Після глобального адаптивного шару усереднення згорткова нейронна мережа повністю збігається з багатошаровим перцептроном — як шарами, так і способом навчання. Цього цілком достатньо для виконання задачі класифікації образів, але для розпізнавання облич підходів замало. Нажаль, якщо залишити архітектуру у такому вигляді, то ми не зможемо досягти якісного розпізнавання.

2. Навчання подібності (підхід ArcFace). Незважаючи на простоту, цей метод виявився надзвичайно ефективним у задачі розпізнавання облич, а тому і дістав відповідну назву. Він пропонує новий спосіб підрахунку ймовірностей класів під час тренувального процесу [7].

Припустимо, у нас є зображення, яке належить до класу J, а всього класів N. Цей підхід не залежить від згорткової нейронної мережі, головне — мати змогу діставати базовий вектор із зображення, позначимо його $x \in R^d$. Тоді останній повнозв'язний шар буде мати лише вагу W розмірності (d, N), зсуву у нього немає. Розглянемо алгоритм присвоєння ймовірностей класам. Перш за все базовий вектор нормалізується:

$$x^{norm} = \frac{x}{\|x\|_2}.$$

Це робиться для того, щоб вивести базовий вектор на гіперсферу у R^d просторі з радіусом 1. Після цього нормалізуємо також матрицю вагів W по стовпцях та множимо базовий вектор на матрицю вагів:

$$W_j^{norm} = \frac{W_j}{\| W_j \|_2} \; \forall j = 1, ..., N,$$

$$Logits = x^{norm} W^{norm},$$

де *Logits* — вектор розмірністю (1, N).

Фактично

$$Logits_j = (x^{norm}, W_j^{norm}) = \frac{(x^{norm}, W_j^{norm})}{\| x^{norm} \|_2 \cdot \| W_j^{norm} \|_2} =$$

$$= \mathrm{Cos}(x^{norm}, W_j^{norm}) \; \forall j = 1, ..., N,$$

тобто *Logits* — це вектор, який зберігає косинусну подібність між x^{norm} та W_j^{norm}. Тоді W_j^{norm} можна вважити деяким середнім вектором для класу J, його ще називають якорем цього класу. Внаслідок навчання матриця W^{norm} буде зберігати оптимальні якорі кожного класу, з якими щоразу буде порівнюватись базовий вектор.

Тепер потрібно дістати кут між базовим вектором, який видала наша нейронна мережа, та якорем справжнього класу J (це робиться лише під час навчання, оскільки в реальному часі ця інформація невідома):

$$Logits_J = \mathrm{Cos}(x^{norm}, W_J^{norm}) = \cos(\theta_J),$$

$$\theta_J = \arccos(\mathrm{Cos}(x^{norm}, W_J^{norm})) = \arccos(Logits_J),$$

θ_J — кут між нашим базовим вектором та справжнім класом j. Після цього додаємо до кута деяку величину m: $\theta_J = \theta_J + m$. Відповідно зменшиться і $\mathrm{Cos}(x^{norm}, W_J^{norm})$. В результаті виправлений вектор *Logits*:

$$\begin{cases} Logits_j = \cos(\theta_j) & \forall j \neq J, \\ Logits_J = \cos(\theta_J + m). \end{cases}$$

Домножуємо кожне значення цього вектора на s, таким чином збільшуючи розмір гіперсфери, на якій розташовані базові вектори:

$$Logits_j = s \cdot Logits_j,$$

вважаємо, що ймовірності кожного класу

$$Output_j = softmax(Logits_j)\ \forall j = 1, ..., N.$$

Далі навчання нічим не відрізняється від навчання будь-якої згорткової нейронної мережі: використовуємо ентропію як функцію втрат та навчаємо мережу методом зворотного поширення помилки.

Тепер детально розглянемо, чому цей метод настільки ефективний. Перш за все оперуємо на одиничній гіперсфері, що дозволяє нам дивитись лише на кути між векторами. Ненормалізованими ймовірностями для наших класів служать косинуси кутів між базовим вектором і якорем кожного з класів. Фактично визначаємо, до якого класу кут найменший, тобто до якого якоря наш базовий вектор знаходиться найближче на одиничній гіперсфері. Але під час навчання збільшуємо кут між базовим вектором нашої мережі та якорем справжнього класу. Це зменшує ймовірність справжнього класу і збільшує функцію втрат. Завдяки цьому мережа завжди буде отримувати градієнт для навчання, це стимулюватиме її якнайдалі відкинути негативні приклади від позитивного якоря, і навпаки. Навіть якщо мережа для зображення даватиме базовий вектор, який матиме кут $\theta_J = 0$ з правильним класом, то все одно збільшуємо цей кут на m. Також множимо наші ненормалізовані ймовірності на s, щоб при великій кількості класів можна було отримати велику ймовірність для правильного класу.

Цей підхід практично не додає складності в обчисленнях, але дозволяє успішно навчати моделі розпізнавання облич і отримувати точності, недосяжні без підходу навчання подібності. У наступному розділі порівняємо результати з методом ArcFace та без нього.

3. Методи оптимізаціх нейронних мереж. Повернемось до нейронних мереж. У цьому випадку функція, яку ми мінімізуємо, — це функція втрат для всього тренувального набору даних, а отже, це сума функції втрат для кожного окремого спостереження:

$$F(x, \theta) = \sum_{i=1}^{N} L(f(x^{(i)}, \theta), y^{(i)}),$$

де $L(f(x^{(i)}, \theta), y^{(i)})$ — функція втрат між передбаченням нейронної мережі та істинним класом для i-го спостереження. Тоді для нейронної мережі стохастичний градієнтний спуск з розміром міні-партії M виглядатиме так:

$$\theta_{t+1} = \theta_t - \alpha_t \cdot \nabla_\theta \sum_{i=1}^{M} L(f(x^{(i)}, \theta), y^{(i)}).$$

Оскільки кожен градієнт міні-партії лише апроксимує істинний градієнт, то при досить малому значенні спостережень у міні-партіях дисперсія градієнту може бути великою. Це приводить до поганої збіжності. Проте цього можна позбутись за допомогою експоненційного зглажування градієнтів, який називатимемо моментом градієнту. Стохастичний градієнтний спуск з моментом градієнту виглядатиме так:

$$h_t = \beta \cdot h_{t-1} + (1-\beta) \cdot \nabla_\theta \sum_{i=1}^{M} L(f(x^{(i)}, \theta), y^{(i)}),$$

$$\theta_{t+1} = \theta_t - \alpha_t \cdot h_t.$$

Адаптивними методами оптимізації називаються методи, які корегують темп навчання для кожного параметра окремо. Це збільшує швидкість збіжності, а також покращує якість знайденого локального мінімуму.

Розглянемо алгоритм Adam, який став дуже популярним для згорткових нейронних мереж [8]:

$$g_t = \nabla_\theta \cdot \sum_{i=1}^{M} L(f(x^{(i)}, \theta), y^{(i)}),$$

$$h_t = h_{t-1} + \beta_1 \cdot g_t,$$

$$v_t = v_{t-1} + \beta_2 \cdot g_t^2,$$

$$\hat{h}_t = \frac{h_t}{1-\beta_1},$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2},$$

$$\theta_{t+1} = \theta_t - \frac{\alpha_t \cdot \hat{h}_t}{\sqrt{\hat{v}_t + \varepsilon}}.$$

Таким чином, для кожного параметра темп навчання буде корегуватись і дорівнювати $\frac{\alpha}{\sqrt{\hat{v}_t + \varepsilon}}$. Тут ε у знаменнику — числова стабільность під час навчання. Цей метод у багатьох задач показує кращі результати, ніж простий метод стохастичного градієнтного спуску.

Результати

Для порівняння різних підходів розпізнавання облич ми обрали набір даних VGGFace2 [9]. Він складається з 3,31 млн зображень 9131 людини. Його створили за допомогою зображень з пошукової системи Google. Спочатку застосовувалися попередньо навчені нейронні мережі для виділення певної кількості найбільш можливих фотографій людини. Після цього зображення перевірялись розмітниками для фінальної ідентифікації. Цей набір даних має такі переваги.

1. Він досить об’ємний, щоб можна було навчати на ньому нейронні мережі різних розмірів без значного перенавчання.

2. Великою проблемою інших наборів даних є мала кількість зображень на кожну людину, проте тут на кожну людину в середньому припадає 363 зображення.

3. Зображення кожної людини взяті в абсолютно різний час та у різних місцях. У деяких людей у наборі є навіть їх дитячі фотографії.

Для валідаційної вибірки ми відклали 50 зображень 500 людей, в загальному 25000 зображень.

Кожне зображення для навчання та валідації попередньо було оброблене (рис. 4).

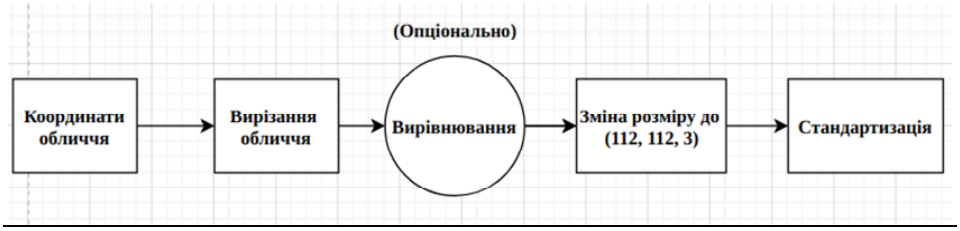

Рис. 4

Під вирівнюванням мається на увазі перетворення фото обличчя таким чином, щоб ключові точки (очі, ніс, рот) не було зміщено. Ця операція, характерна для будь-якого алгоритму, покращує процес розпізнавання.

Стандартизація тут — це віднімання від зображення вектору [0,5; 0,5; 0,5] та ділення на [0,5; 0,5; 0,5].

Основна метрика у нас — Accuracy, тобто точність. Це зумовлено тим, що валідаційна вибірка повністю збалансована. Також спостерігатимемо за метрикою Accuracy@5 (кількість разів, коли реальна людина знаходиться у топ-5 наших передбачень).

Практично всі експерименти проводилися ітеративно, тобто, обравши найкращий підхід у попередньому етапі (найкращий оптимізатор), ми використовували вже його і далі перевіряли, наприклад, архітектуру мережі. Це називається жадібним алгоритмом вирішення задачі. Він не гарантує пошук глобального мінімуму, проте дуже швидкий.

Щоб зробити певний експеримент, від якого будемо відштовхуватись, використовуватимемо такі параметри мережі: архітектура: EfficientNet-B0; без підходу навчання подібності; попередня обробка зображення: просте вирізання обличчя (без вирівнювання), зміна розміру до (112, 112, 3), стандартизація; розмір міні-партії: 930 зображень; кількість епох: 50; оптимізатор: стохастичний градієнтний спуск з темпом навчання $\alpha_0 = 0$. Результати: Accuracy = 0,665, Accuracy@5 = 0,703. Фактично кожне третє обличчя з валідаційного набору даних ідентифіковано невірно.

Тепер візьмемо попередній експеримент і просто додамо навчання подібності, а саме, ArcFace з базовими параметрами s = 64 та m = 0,2, залишивши усі інші параметри сталими. Результати: Accuracy = 0,7422, Accuracy@5 = 0,826. Бачимо, що просто додавши концепцію навчання подібності, ми отримали великий приріст, а саме, 8 % точності. Проте мережа досі далека від оптимального значення метрики, тому що базові параметри не завжди добре працюють на інших наборах даних.

Спробуємо додати вирівнювання обличчя замість простого вирізання. Результати: Accuracy = 0.758, Accuracy@5 = 0.854. Отже, така проста попередня обробка даних збільшує точність понад 1 % та приблизно 3 % у Accuracy@5, тому обов'язково додаємо її до нашого фінального алгоритму навчання.

Великі розміри міні-партій не завжди приносять користь. Менші міні-партії вносять шум до градієнтів, і це виступає певною регуляризацією для мережі. Тому спробуємо змінити розмір міні-партії з 930 до 540, а також $\alpha_0 = 0{,}1$. Результати: Accuracy = 0,802, Accuracy@5 = 0,895. Приріст в 4 % точності — це великий приріст, тому можемо зробити висновок, що правильний підбір розміру міні-партії є критичним.

Результати цього блоку експериментів показані у табл. 1.

Таблиця 1

Arcface s=64, m=0,2	Вирівнювання обличчя	Розмір міні-партії 540	Accuracy	Accuracy@5
–	–	–	0,665	0,703
+	–	–	0,7422	0,826
+	+	–	0,758	0,854
+	+	+	**0,815**	**0,899**

На даному етапі ми знайшли задовільні параметри, щоб почати ширший пошук інших параметрів: архітектура: EfficientNet-B0; ArcFace з параметрами s = 64 та m = 0,2; попередня обробка зображення: вирізання обличчя з вирівнюванням, зміна розміру до (112, 112, 3), стандартизація; розмір міні-партії: 540 зображень; кількість епох: 50; оптимізатор: стохастичний градієнтний спуск з темпом навчання $\alpha_0 = 0{,}1$.

Спробуємо підібрати параметри для ArcFace, оскільки вони одні з найважливіших параметрів у всьому навчанні. Спочатку змінимо $m = 0{,}2$ на $m = 0{,}3$. Результати: Accuracy = 0,781, Accuracy@5 = 0,83. Результати набагато гірші. Це означає, що більший штраф для кута до правильного класу не завжди покращує результат, важливо знайти золоту середину, тому зупинимось на $m = 0{,}2$.

Поекспереметуємо з параметром s мережі. Результати: при $s = 32$: Accuracy = = 0,858, Accuracy@5 = 0,928. Результати: при s = 24: Accuracy = 0,876, Accuracy@5 = = 0,933. Результати: при $s = 16$: Accuracy = 0,862, Accuracy@5 = 0,93. Бачимо, що оптимальним варіантом є $s = 24$, що в 2,6 рази менше від базового значення. Він збільшує точність понад 6 %, що є значним покращенням.

Не зважаючи на теоретичну обґрунтованість адаптивних методів оптимізації, для задачі розпізнавання облич вони виявились гіршими за простий стохастичний градієнтний спуск. Для методу Adam отримано такі результати: Accuracy = 0,612, Accuracy@5 = 0,771.

Результати цього блоку експериментів знаходяться у табл. 2.

Таблиця 2

ArcFace				Adam	Accuracy	Accuracy@5
$m=0{,}3$	$s=32$	$s=24$	$s=16$			
+	–	–	–	–	0,781	0,83
–	+	–	–	–	0,858	0,928
–	–	+	–	–	**0,876**	**0,933**
–	–	–	+	–	0,862	0,93
–	–	–	–	+	0,612	0,771

Отже, з цього блоку досліджень ми знайшли оптимальний параметр s для ArcFace. Оптимізатором залишаємо стохастичний градієнтний спуск. Останній блок експериментів присвячений підбору архітектури.

Для архітектури EfficienNet-B3 ми оримали такі результати: Accuracy = 0,901, Accuracy@5 = 0,945, а для архітектури SE-ResNet50: Accuracy = 0,92, Accuracy@5 = = 0,951. Бачимо, що архітектура з механізмом уваги і більшою кількістю шарів проявила себе набагато краще на задачі розпізнавання облич.

Результати останнього блоку експериментів знаходяться у табл. 3.

Таблиця 3

Архітектура		Accuracy	Accuracy@5
EffNet-B3	Se-Resnet50		
–	–	0,881	0,941
+	–	0,901	0,945
–	+	**0,92**	**0,951**

Отже, виходячи з наших експериментів, найкраща конфігурація для навчання моделі розпізнавання облич така:

1) архітектура — SE-ResNet50;

2) ArcFace з параметрами $s=24$ та $m=0{,}2$;

3) попередня обробка зображення: вирізання та вирівнювання обличчя, стандартизація;

4) 80 епох навчання;

5) міні-партія розміром 540;

6) оптимізатор — стохастичний градієнтний спуск з початковим темпом навчання $\alpha_0 = 0{,}1$.

Висновки

У даній публікації ми перевірили ряд гіпотез щодо покращення якості роботи мережі для розпізнавання облич. У більшості випадків підходи, які в теорії мали давати кращі результати, дійсно це робили.

Зокрема, навчання подібності є критичним для задачі розпізнавання облич. Без цього підходу задача невирішувана. Дуже важливо підбирати параметри для методу ArcFace щодо свого набору даних, тому що певним базовим варіантом є $s = 64$ та $m = 0{,}2$, а фактично найкращим варіантом виявились значення $s = 24$ та $m = 0{,}2$.

Наступний важливий фактор — підбір архітектури моделі, достатньо місткої, щоб добре класифікувати понад 9000 класів. На практиці важливо використовувати більш глибоку мережу з механізмом уваги.

Не менш важливим є оптимізатор. Адаптивні методи оптимізації себе не виправдали і на практиці показали погані результати. Тому необхідно пробувати і адаптивні методи типу Adam або RAdam, і простий стохастичний градієнтний спуск.

Загалом, використовуючи всі вищеперелічені підходи, ми змогли отримати точність 92 % на досить складному наборі даних, що на 25,5 % краще за базовий експеримент.

А.М. Літвинчук, Л.В. Барановська

ПОКРАЩЕННЯ МОДЕЛЕЙ РОЗПІЗНАВАННЯ ОБЛИЧ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ, НАВЧАННЯ ПОДІБНОСТІ ТА МЕТОДІВ ОПТИМІЗАЦІЇ

Розпізнавання облич — це одна з основних задач комп'ютерного зору. Вона має безліч прикладних застосувань, що призвело до величезної кількості досліджень у цій сфері. І хоча дослідження відбувались з початку розвитку комп'ютерного зору, адекватних результатів змогли досягнути лише за допомогою згорткових нейронних мереж. У даній роботі проведено порівняльний аналіз методів розпізнавання облич до згорткових нейронних мереж. Розглянуто набір архітектур нейронних мереж, методів навчання подібності та оптимізації. Проведено ряд експериментів, виконано порівняльний аналіз розглянутих методів покращення згорткових нейронних мереж, в результаті отримано універсальний алгоритм для навчання моделі розпізнавання облич. Для порівняння різних підходів розпізнавання облич ми обрали набір даних VGGFace2. Він складається з 3,31 млн зображень 9131 людини. Його створили за допомогою зображень з пошукової системи Google. Спочатку застосовувалися попередньо навчені нейронні мережі для виділення певної кількості найбільш можливих фотографій людини. Після цього зображення перевірялись розмітниками для фінальної ідентифікації. Для валідаційної вибірки відклали 50 зображень 500 людей, загалом 25000 зображень. Практично всі експерименти проводилися ітеративно. Тобто, обравши найкращий підхід у попередньому етапі (наприклад, найкращий оптимізатор), ми використовували вже його і далі перевіряли, наприклад, архітектуру мережі. Як і очікувалось, нейронні мережі з більшою кількістю параметрів та складнішою архітектурою показували кращі результати у наведеній в роботі задачі. Серед розглянутих нами моделей найкращою виявилась Se-ResNet50. Навчання подібності — це метод, за допомогою якого можливо досягнути хорошої точності. Без цього методу задачу вирішити було б неможливо. Для оптимізації нейронних мереж ми розглядали і адаптивні, і прості оптимізатори. Як показано у роботі, для даної задачі найкращим виявився стохастичний градієнтний спуск з моментом, а адаптивні методи показали поганий результат. Загалом, використовуючи різні підходи, ми змогли отримати точність 92 % на досить складному наборі даних, що на 25,5 % краще за базовий експеримент. Подальший розвиток даного дослідження можливий завдяки покращенню архітектури нейронної мережі, збору більшої кількості даних та застосуванню кращих методів регуляризації.

Ключові слова: згорткові нейронні мережі, розпізнавання облич, навчання подібності, методи оптимізації.

A.M. Litvynchuk, L.V. Baranovska

IMPROVING FACE RECOGNITION MODELS USING CONVOLUTIONAL NEURAL NETWORKS, METRIC LEARNING AND OPTIMIZATION METHODS

Face recognition is one of the main tasks of computer vision. It has many applications, which has led to a huge amount of research in this area. And although research in the field has been going on since the beginning of the computer vision, good results could be achieved only with the help of convolutional neural networks. In this work, a comparative analysis of facial recognition methods before convolutional neural networks was performed. A set of neural network architectures, methods of metric learning and optimization are considered. There were performed bunch of experiments and comparative analysis of the considered methods of improvement of convolutional neural networks. As a result a universal algorithm for training the face recognition model was obtained. To compare different approaches of face recognition, we chose a dataset called VGGFace2. It consists of 3,31 million images of 9131 people. It was created using images from the Google search engine. Initially, pre-trained neural networks were used to select photographs with humans. The images were then checked mannualy. For the validation sample, we set aside 50 images of 500 people, for a total of 25,000 images. Almost all experiments were performed iteratively. For example, we choose the best optimizer and then we use it to search for best arctitecture. As expected, neural networks with more parameters and more sophisticated architecture showed better results in this task. Among the considered models the best was Se-ResNet50. Metric learning is a method by which it is possible to achieve good accuracy in face recognition. Without this method it would be impossible to solve the problem. To optimize neural networks, we considered both adaptive and simple optimizers. It turned out that the stochastic gradient descent with moment is the best for this problem, and adaptive methods showed a rather poor result. In general, using different approaches, we were able to obtain an accuracy of 92 %, which is 25,5 % better than the baseline experiment. We see next ways for the further development of the research subject: improving neural network architecture, collecting more data and applying better regularization techniques.

Keywords: convolutional neural networks, face recognition, metric learning, optimization methods.

1. Zhao W., Chellapa R. Image-based face recognition: issues and methods. *Conference on Computer Vision and Pattern Recognition*. 2002. URL: https://www.semanticscholar.org/paper/Image-based-Face-Recognition%3A-Issues-and-Methods-1-Zhao-Chellappa/dbd5e9691cab2c515b50dda3d0832bea6eef79f2.
2. Tate G. ResNet-50 — a misleading machine learning inference benchmark for megapixel images. 2019. URL: https://www.eenewseurope.com/news/resnet-50-misleading-machine-learning-inference-benchmark-megapixel-images (дата звернення: 06.08.2021).
3. He K., Zhang X., Ren S., Sun J., Deep residual learning for image recognition. *Conference on Computer Vision and Pattern Recognition*. 2015. URL: https://www.semanticscholar.org/search?q=3.%09He%20K.%2C%20Zhang%20X.%2C%20Ren%20S.%2C%20Sun%20J.%2C%20Deep%20Residual%20Learning%20for%20Image%20Recognition.%20Conference%20on%20Computer%20Vision%20and%20Pattern%20Recognition.%202015.&sort=relevance.
4. Hu J., Shen L., Albanie S. Squeeze-and-Excitation Networks. *Conference on Computer Vision and Pattern Recognition*. 2018. P. 7132–7141.
5. EfficientNet: как масштабировать нейросеть с использованием AutoML. 2019. URL: https://neurohive.io/ru/novosti/efficientnet-kak-masshtabirovat-nejroseti-s-pomoshhju-automl/ (дата звернення: 05.05.2020).
6. Le Q., Tan M. EfficientNet: rethinking model scaling for convolutional neural networks. International Conference on Machine Learning. 2019. **36**. P. 6105–6114.
7. Deng J., Guo J., Zafeiriou S., Xue N. ArcFace: additive angular margin loss for deep face recognition. *Conference on Computer Vision and Pattern Recognition*. 2018. P. 4685–4694.
8. Kingma D. P., Ba J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*. 2014. URL: https://www.semanticscholar.org/paper/Adam%3A-A-Method-for-Stochastic-Optimization-Kingma-Ba/a6cb366736791bcccc5c8639de5a8f9636bf87e8.
9. Cao Q., Shen L., Xie W. VGGFace2: A dataset for recognising faces across pose and age. *Conference on Computer Vision and Pattern Recognition*. 2017. P. 67–74.

Отримано 07.08.2021