УДК 519.6 *http://dx.doi.org/10.30970/vam.2024.32.12028*

РОЗРОБКА ТА ОПТИМІЗАЦІЯ АНТИСПУФІНГОВОЇ НЕЙРОННОЇ МЕРЕЖІ

І. Смоловик, Ю. Щербина

Львівський національний університет імені Івана Франка,
вул. Університетська 1, Львів, 79000, Україна
e-mail: i.m.smolovyk@gmail.com, yuriy.shcherbyna@lnu.edu.ua

Розглянуто високоточні методи побудови антиспуфінгових згорткових нейронних мереж, які забезпечують ефективне розпізнавання штучно згенерованих облич на основі різних метрик, проаналізовано існуючі підходи до побудови мереж, що вирішують згадану проблему та відзначено їхні переваги та недоліки, досліджено наслідки впливу різними оптимізаторами поодинці так комбіновано на якість роботи мережі, на підставі отриманого аналізу створено власну антиспуфінгову модель на базі згорткової нейронної мережі, як однієї з найточніших, для досягнення бажаної ефективності виконання та точності.

Ключові слова: антиспуфінг, згорткова нейронна мережа, розпізнавання облич, точність.

1. Вступ

Останніми роками надзвичайного розвитку зазнала галузь розробки штучних нейронних мереж, що вирішують питання розпізнавання облич. Оскільки кількість сфер застосування цих технологій постійно збільшується, то це зумовило створення великої кількості датасетів і можливості тренувати нові нейронні мережі на заздалегідь підготовлених великих кількостях даних, що також забезпечує надзвичайну точність обчислень і результатів, розв'язання задачі розпізнавання облич гарантує ідентифікацію, забезпечення необхідного рівня захисту персональних даних, зокрема біометрії, а також відкидає можливість проникнення третіх осіб до структур із обмеженим доступом, тобто такі, що на різних рівнях потребують захисту процесів, що відбуваються в згаданих структурах, і даних, що можуть зберігатись там. Втім, поруч із новітніми системами ідентифікації облич з'явилися і так звані спуфінгові, що означає мережі для генерування штучних облич. Такі мережі широко використовують для атак і підробки біометричних даних, тому на противагу їм проводили тривалі дослідження та розробляли алгоритми для створення антиспуфінгових нейронних мереж, здатних відстежувати підробки на основі різних метрик. Одним із найбільш ефективних видів глибоких моделей, зданих вирішувати задану проблему, вважаються згорткові нейронні мережі. Вони мають багато переваг, на зразок можливості обробляти дані у початковому стані, тобто без додаткових маніпуляцій і високої здатності до навчання.

2. Згорткові нейронні мережі

Згорткові нейронні мережі (ЗНМ, англ. *convolutional neural networks*, CNN) – це підвид глибоких штучних нейронних мереж прямого поширення, що головно застосовується до розпізнавання об'єктів і класифікації зображень. Обробка й сприйняття даних у згорткових нейронних мережах відбувається у вигляді тензорів,

© Смоловик І., Щербина Ю., 2024

отже, забезпечується можливість працювати з даними зображень у їх природній формі. Для задач розпізнавання в умовах використання великих об'ємів даних потрібно, щоб модель мала високу здатність до навчання та великий відсоток правильних припущень щодо ознак зображення. Порівняно з традиційними нейронними мережами прямого поширення зі схожою кількістю шарів, згорткові нейронні мережі мають вищу здатність до навчання, бо містять набагато менше параметрів і зв'язків.

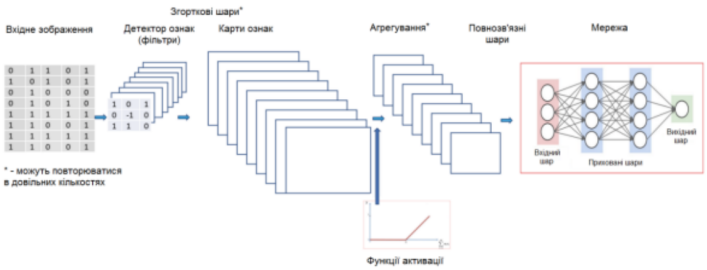

Рис. 1. Архітектура ЗНМ
CNN Architecture

Структура згорткової нейронної мережі відповідає загальноприйнятій структурі нейронної мережі. Мережа складається з вхідного шару, певної кількості прихованих шарів s вихідного шару. Приховані шари зазвичай складаються зі згорткових шарів, шарів агрегування (субдискретизації), нормалізуючих та повнозв'язних шарів. Ці шари пов'язані між собою шарами з визначеними активаційними функціями.

2.1. Згорткові шари

Операції згортки виконують у згорткових шарах нейронної мережі та є найголовнішим їхнім елементом. Саме згортка є математичною операцією, яка за допомогою дії ядра згортки (детектор ознак, англ. *kernel або feature detector*) k на вхідне зображення у вигляді тензора x отримує як результат карту ознак m (англ. textitfeature map). Кожний детектор ознак складається з деякої визначеної кількості фільтрів (англ. textitfilters), кожний з яких nfrj; є тензором параметрів з розмірністю, яка зазвичай дає змогу покривати лише невелику частину зображення в ширину та висоту, але обов'язково має відповідати розмірності в глибину. Такі фільтри і є параметричною частиною нейронної мережі, до якої застосовується навчання. Загалом вигляді операція дискретної згортки може бути записана математично так:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a).$$

Значення кожного детектора ознак за межами визначеної області його дії вважаються нульовими, тому нескінченну суму можна замінити на скінченну зі значеннями в заданій області. Використовуючи властивість комутативності для операції математичної згортки, отримуємо спрощену функцію, придатну для застосування в нейронних мережах і створення детекторів для згорткового шару, та яка споріднена

з взаємокореляційною функцією для значень зображення

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n).$$

Оскільки операція згортки комутативна, то формулу можна записати так:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n),$$

де $S(i,j)$ – елемент карти ознак з координатами i та j; I – вхідне зображення, K – детектор ознак; m, n – розмірності детектора ознак. Приклад фільтра для операції згортки зображений на рис. 2.

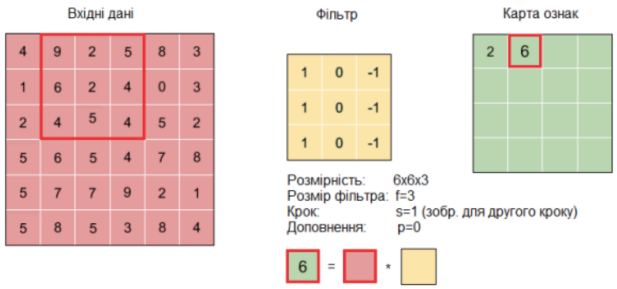

Рис. 2. Фільтр для операції згортки
Filter for convolution operation

2.2. Агрегувальні шари

Функція пулінгу замінює вихід мережі в деякій точці зведеної статистики близьких виходів. У будь-якому випадку пулінг дає змогу зробити уявлення приблизно інваріантним щодо малих паралельних переносів входу, це означає, що якщо зрушити вхід на невелику величину, то значення більшості виходів, що піддаються впливу пулінгу зміняться.

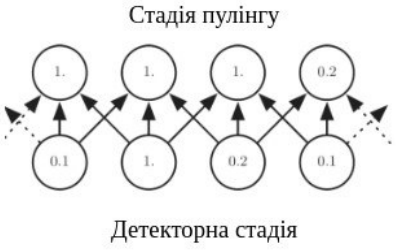

Рис. 3. Механізм агрегування
Aggregation mechanism

На рис. 3 показано приклад роботи цього механізму. Локальна інваріантність корисна, якщо нас більше цікавить сам факт існування певної ознаки, а не її точне

розташування. У більшості завдань пулінг необхідний для обробки входів змінного розміру.

2.3. Розріджена взаємодія

На відміну від звичних нейронних мереж, у котрих на кожному шарі застосовується множення на матрицю параметрів, тобто кожен вихідний блок взаємодіє з кожним вхідним блоком, у згорткових нейронних мережах взаємодія є розрідженою, внаслідок того, що ядро менше за вхід. Візьмемо, наприклад, зображення, яке міститиме надзвичайно велику кількість пікселів, проте насправді визначними з них будуть лише деякі, які окреслюють контури об'єкта, а отже, досягається ефективність в обсягу пам'яті, бо запам'ятовувати потрібно лише малу частину параметрів.

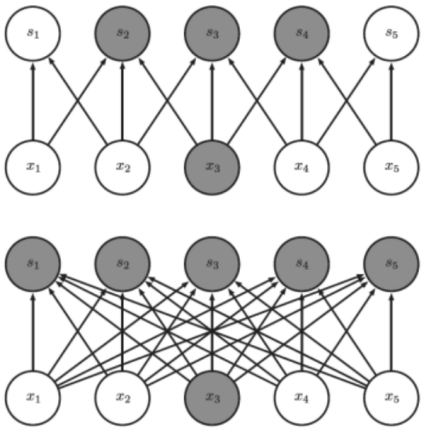

Рис. 4. Розріджена зв'язність
Sparse connectivity

2.4. Залишкова мережа (Residual Network)

Часто задля досягнення вищої точності роботи моделі додають додаткову, велику кількість шарів, що в багатьох випадках і справді збільшує ефективність, проте часом може призводити до такої проблеми: глибокого як навчання, як зникнення чи розрив градієнта, тобто градієнт набуває значення близьке до 0 чи навпаки надзичайно велике. Зобразимо на прикладі графіку (рис. 5)

Отже, бачимо, що точність моделі з 20-ма шарами помітно вища за 56-шарову.

Для вирішення цієї проблеми були створені залишкові нейронні мережі. Головна ідея такої моделі полягає в ідеї залишкових блоків, використовується техніка під назвою пропуск з'єднань. Фактично, це з'єднання пропуску поєднує активації шару з наступними шарами, пропускаючи деякі шари між ними. Це й утворює залишковий блок. Resnets будують шляхом складання цих залишкових блоків разом. Перевага додавання цього типу з'єднання пропуску полягає в тому, що якщо будь-який рівень негативно впливатиме на продуктивность всієї архітектури, тоді його буде пропущено завдяки регуляризації. Отже, це призводить до навчання

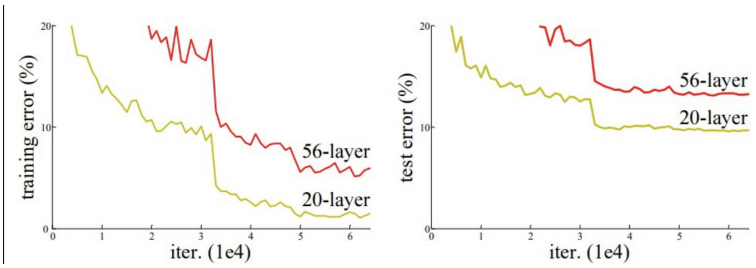

Рис. 5. Порівняння роботи моделей з різною кількістю шарів
Comparing the performance of models with different numbers of layers

дуже глибокої нейронної мережі без проблем, зумовлених зникненням/розривом градієнта.

3. Антиспуфінг на базі ЗНМ, існуючі підходи до розробки

Антиспуфінг облич – це технологія, яку використовують для виявлення спроб підробки або обману у разі аутентифікації або ідентифікації особи на основі фотографій або відео її обличчя. Основна мета цієї технології – запобігати псевдоаутентифікації, коли спроби фальсифікації обличчя або фотографій використовують для незаконного доступу до системи або проникнення в захищену область. Антиспуфінг гарантує, що використовуються лише реальні обличчя для аутентифікації.

Основні завдання та характеристики антиспуфінгу охоплюють: виявлення підроблених фотографій і відео, аналіз текстурних і геометричних характеристик, використання біометричних особливостей, використання глибокого навчання, застосування до систем біометричної аутентифікації. Уразі створення антиспуфінгових нейронних мереж на базі згорткових нейронних мереж (CNN) існує кілька основних підходів і методів, які варто розглянути детальніше.

– Підхід на основі патчів (Patching-Based Approach).

У цьому підході вхідне зображення розбивається на невеликі частини або "патчі", і кожен патч аналізується окремо за допомогою CNN. Використання патчів дає змогу виконувати детальний аналіз текстур і геометрії окремих частин обличчя та визначати, чи є якісь аномалії або різниці між ними та нативними зразками обличчя.

– Аналіз в часі (Temporal Analysis).

Використовують відеодані для аналізу рухів і змін обличчя в часі. Модель аналізує послідовність кадрів відео та виявляє нелогічні зміни в обличчі. Наприклад, спроби підроблення можуть супроводжуватися швидкими змінами обличчя, які незвичайні для реальних ситуацій.

– Підхід на основі глибини (Depth-Based Approach).

У цьому підході використовують дані про глибину, отримані за допомогою 3D-камер або інших датчиків. Аналіз глибини допомагає виявляти аномалії в геометрії обличчя, такі як відсутність глибини, незвичайні форми обличчя або розміри, що відрізняються від норми.

– Спектральний аналіз (Spectral Analysis).

Спектральний аналіз охоплює аналіз характеристик спектрів зображення, таких як частоти або спектральні особливості. Аномалії в спектрі можуть засвідчувати підроблення, оскільки підроблені зображення можуть мати інші спектральні характеристики, ніж нативні.

– Глибоке навчання (Deep Learning).

Глибоке навчання використовує згорткові нейронні мережі (CNN), щоб вивчити ознаки та шаблони, які свідчать про підроблення. Модель навчається на великих наборах даних, щоб автоматично виявляти антиспуфінг. Може використовуватися в комбінації з іншими підходами.

– Багатомодальний підхід (Multi-Modal Approach).

Багатомодальний підхід охоплює в себе використання кількох методів і датчиків для виявлення підроблень. Наприклад, він може комбінувати аналіз зображень, відео, 3D-даних та інших біометричних характеристик для поліпшення точності виявлення антиспуфінгу.

– Адверсарне навчання (Adversarial Training).

Адверсарне навчання це тренування моделей з урахуванням атак, спрямованих на підробку. Модель навчається відстоюватися від атак і виявляти спроби підроблення обличчя, навчаючись розрізняти між підробленими та реальними зображеннями.

4. Проєктування та робота з моделлю

Уся подальша робта з моделлю напряму залежить від вигляду, розміру та формату датасету, який використовуватиметься для навчання цієї моделі. Було обрано датасет NUAA, що місить в загальній кількості 12614 зображень облич, датасет поділений на дві групи, за якими фактично і буде відбуватися класифікація, а саме: реальні обличчя та фальсифіковані. Кожна з груп також містить папки з зображеннями різних людей, при чому в кожній з папок фотографії лише однієї людини, де фотографії різняться за певним приницпом: у кожній папці обов'язково є зображення в окулярах, без окулярів, з поворотом голови вправо та вліво, нахилами голови, з різними умовами освітлення, включно з природним світлом і штучним, або їх відсутністю. Підібраний датасет чудово задовольняє умови сформульованої задачі та дає високу можливість до навчання новоствореній моделі. Усі зображення датасету за попередньої обробки перед навчанням моделі зводяться до одного розміру, що становить 64*64.

4.1. Побудова та тренування моделі

Для проєктування, тренування та тестування мережі було використано такі інструменти: pytorch та openCV. Певні алгоритми, такі як LBP, реалізовуємо вручну. Перед початком роботи з моделюю усю задану вибірку було поділено на тренувальні та тестові дані, оскільки в датасеті NUAA фейкові та реальні зображення розділені, то їх довелося перемішати між собою.

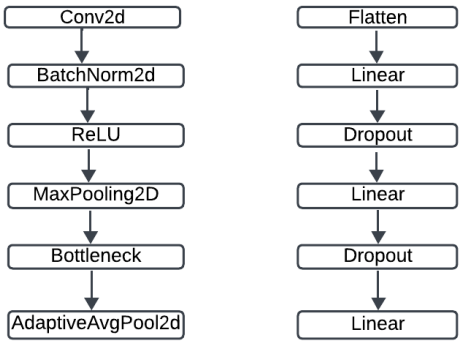

Рис. 6. Архітектура моделі
Model architecture

4.1.1. Архітектура моделі

Архітектура моделі зображена на рис. 6. Побудована модель складається зі згорткового шару з 64 фільтрами розміром 7x7 і з кроком 2x2. Цей шар допомагає вилучити низку ознак з вхідних зображень, функції активації ReLU, згорткового блока (cnn), який розпочинається з BatchNorm2d для нормалізації даних, далі застосовується функція активації ReLU, використовується MaxPool2d для пулінгу даних з ядром розміром 3x3 і кроком 2x2.

Після згорткового блока в мережі є послідовність блоків Bottleneck, кожен з яких має таку структуру: згортковий шар Conv2d розміром 1x1 для зменшення кількості каналів BatchNorm2d для нормалізації, згортковий шар Conv2d розміром 3x3 з функцією активації ReLU та подальшою нормалізацією, ще один згортковий шар Conv2d розміром 1x1 для збільшення кількості каналів і знову BatchNorm2d для нормалізації, завершується функцією активації ReLU.

Після Bottleneck блоків використовується AdaptiveAvgPool2d для пулінгу даних до розміру (1, 1), щоб отримати вектор ознак фіксованого розміру, Flatten для перетворення цього вектора в одномірний вектор. Завершують архітектуру мережі повнозв'язні шари, які зменшують розмірність з 256 до 128, потім до 64 та, нарешті, до 2. В міру проходження крізь ці Linear шари також використовується функція активації ReLU, а після кожного з них застоосовують Dropout шари з параметром p=0.5, за допомогою цих шарів можна уникнути перенавчання моделі.

Оглянувши архітектуру моделі в загальному, можна зробити висновок, що ця модель має базову архітектуру ResNet зі згортковими блоками Bottleneck і шарами пулінгу, які використовують для виділення ознак з вхідних зображень, після чого дані передаються через послідовність повністю з'єднаних шарів для класифікації.

4.2. Оптимізація

Для досягнення кращих результатів було обрано комплексний підхід. Спершу вхідні дані передобробили так, що на момент поділу даних на тестову та тренувальну вибірки зображення мали однакові розміри, тоді було застосовано алгоритм LBP

(Local Binary Pattern), основна ідея якого полягає в тому, що для кожного пікселя зображення порівнюється його значення зі значеннями сусідніх пікселів. Якщо значення пікселя більше або дорівнює значенню сусіднього пікселя, то в результуючому шаблоні встановлюється 1, в іншому випадку – 0. Потім для кожного пікселя створюється бінарний шаблон, і ці шаблони використовують для створення гістограми LBPH для кожного об'єкта на зображенні. Наведемо, наприклад, гістограму створену під час опрацювання алгоритмом нашого датасету.

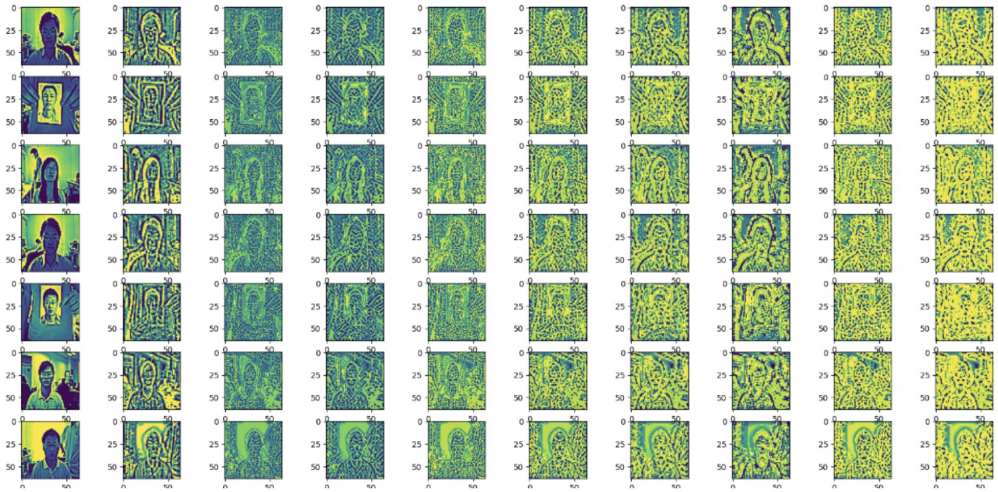

Рис. 7. Гістограма LBP
LBP histogram

LBPH може бути використаний для розпізнавання об'єктів або обличчя на зображеннях, а також для вирізання ознак та опису текстур. Алгоритм має деякі переваги, такі як інваріантність до освітлення та простота в реалізації. Після цього було додано чіткості зображенням за допомогою можливостей бібліотеки openCV.

```
image=cv2.filter2D(image, -1, self.kernel)
```

Застосуємо до моделі оптимізатор adam – один із найефективніших існуючих алгоритмів оптимізації, що поєднує ідеї та методи інших алгоритмів оптимізації, таких як Momentum та RMSprop, і додає до них деякі власні унікальні особливості. Для зручності поділу даних та збільшення ефективності моделі поділимо датасет на батчі, розмір батчу задамо 64, будемо проходитися в межах кожної епохи саме по батчах, що вважається ефективнішим, адже так досягаємо більшої стійкості до шуму, а також збільшуємо швидкість навчання.

5. Аналіз результатів

Для початку перевіримо як працює модель, задамо шлях до зображення та попросимо мережу зробити передбачення справжнє воно, чи фейкове. Запустимо модель і переконаємось, що вона працює правильно, оскільки зображення насправді справжнє.

```
/content/drive/MyDrive/raw/ClientRaw/0010/0010\_01\_05\_03\_121.jpg
```

```
Prediction: REAL
```

Перевіряємо роботу моделі, обчислюючи точність на кожному етапі навчання, а також результуюче остаточне значення. Оскільки використаний інтсрумент pytorch не дає можливості отримувати готове значення точності, то будемо обчислювати її вручну, підраховуючи кількість успішних результатів з валідаційної вибірки на кожному етапі та поділивши отримане значення на загальну кількість запропонованих тестових зображень.

В наслідок проведених досліджень і тренуванні моделі вдалося досягнути точності з оптимізацією близької до 98,5%, а без оптиміації 94,8%. Спостерігаємо помітне поліпшення внаслідок застосування оптимізації.

Для оцінки впливу оптимізації на модель порівняємо графіки втрат для моделі з оптимізацією та без, а також їхні графіки точності на кожній з 11 епох, які пройшла модель у процесі тренування.

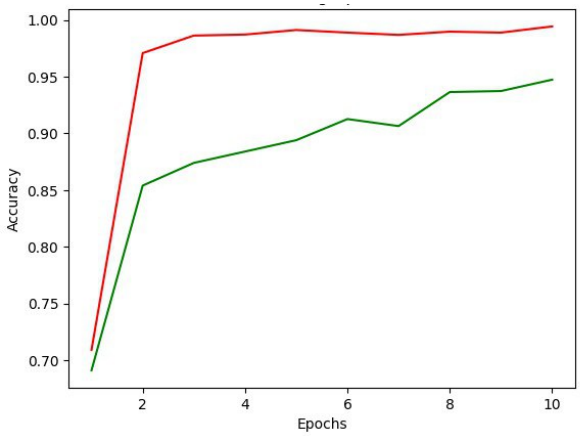

Рис. 8. Зміна точності моделі
Changing the model accuracy

На рис. зображено як змінювалася точність моделі на кожній наступній епосі, тобто повному проході по заданому датасету, загалом таких епох було 10. Зеленою лінією зображено значення для моделі без оптимізації, а червоною – з оптимізацією. Можна зробити висновок, що оптимізація дала свої результати, та допомогла збільшити точність на майже 4%

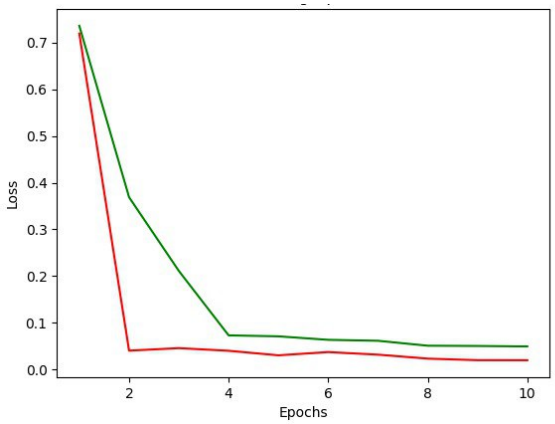

Рис. 9. Грфік втрат
Loss schedule

Такий самий графік було побудовано для порівняння втрат на тренуванні кожної з моделей, на графіку червоною лінією зображено втрати для моделі з оптимізацію, а зеленою – без оптимізації на кожній з епох тренування моделі, як бачимо втрати дещо знизились внаслідок застосування різних технік оптимізації. Фінальне значення втрат для моделі без оптимізації становить 0,04 та 0,02 для моделі з оптимізацією, бо на першій епосі ці значення були суттєво вищими, а саме становили 0,71 та 0,73 для моделей з оптимізацією та без, відповідно.

6. Висновки

Отже, було проведено дослідження та проєктування модифікаованої антиспуфінгової згорткової нейронної мережі, яка фактично вирішує задачу класифікації облич на реальні та фейкові, випробувано різні методи оптимізації та налаштовано їх так, що достягнуто точності роботи мережі 98,5%. Це високі результати, які дає змогу вважати модель вдало побудованою архітектурно та налаштовано так, що найбільше сприяє успішному навчанню та подальшому використанню в цілях авторизації та автентифікації. Така модель стійка до спуфінгових атак і має значний потенціал.

Список використаної літератури

1. Zeiler M.D. Visualizing and understanding convolutional networks / M.D. Zeiler, R. Fergus. – 2014.
2. Goodfellow I. Deep Learning // Ian Goodfellow, Aaron Courville, Yoshua Bengio. – 2015.
3. Girshick R. Region-based convolutional networks for accurate object detection and segmentation / R. Girshick, J. Donahue, T. Darrell, J. Malik. – 2015.
4. Farabet C. Large-scale FPGA-based convolutional networks. In R. Bekkerman M. Bilenko and J. Langford editors / C. Farabet, Y. LeCun, K. Kavukcuoglu, E. Culurciello, B. Martini, P. Akselrod, S. Talay. – Scaling up Machine Learning: Parallel and Distributed Approaches. Cambridge University Press, 2011.
5. Chellapilla K. High Performance Convolutional Neural Networks for Document Processing. In Guy Lorette, editor, Tenth International Workshop on Frontiers in Handwriting Recognition, La Baule (France) / K. Chellapilla, S. Puri, P. Simard. – Universite de Rennes 1, Suvisoft, 2006.

6. Ciresan D. Flexible, High Performance Convolutional Neural Networks for Image Classification / Dan Ciresan, Ueli Meier, Jonathan Masci , Luca M. Gambardella, Jurgen Schmidhuber. – 2011.

Стаття: надійшла до редколегії 21.11.2023
доопрацьована 20.03.2024
прийнята до друку 19.06.2024

DEVELOPMENT AND OPTIMIZATION OF ANTI-SPOOFING NEURAL NETWORK

I. Smolovyk, Yu. Shcherbyna

Ivan Franko National University of Lviv,
1, Universytetska str., 79000, Lviv, Ukraine,
e-mail: i.m.smolovyk@gmail.com, yuriy.shcherbyna@lnu.edu.ua

In the paper high-precision methods of building anti-spoofing convolutional neural networks, which provide effective recognition of artificially generated faces based on various metrics, were considered, existing approaches to building networks that solve the mentioned problem were analyzed and their advantages and disadvantages were noted, the consequences of the influence on the quality of the network of various optimizers both individually and in combined way used were researched. On the basis of the obtained analysis, an own anti-spoofing model was created based on a convolutional neural network, as one of the most accurate, to achieve the desired performance efficiency and accuracy.

Key words: anti-spoofing, convolutional neural network, face recognition, accuracy.