



Politechnika
Śląska

POLITECHNIKA ŚLĄSKA

WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI

Praca inżynierska

Narzędzie do ekstrakcji cech głębokich za pomocą konwolucyjnych
sieci neuronowych

autor: Mikołaj Habarta

kierujący pracą: dr hab. inż. Michał Kawulok

Gliwice, styczeń 2021

Oświadczenie

Wyrażam zgodę / Nie wyrażam zgody* na udostępnienie mojej pracy dyplomowej / rozprawy doktorskiej*.

Gliwice, dnia 9 stycznia 2021

.....
(podpis)

.....
(poświadczenie wiarygodności
podpisu przez Dziekanat)

* podkreślić właściwe

Oświadczenie promotora

Oświadczam, że praca „Narzędzie do ekstrakcji cech głębokich za pomocą konwolucyjnych sieci neuronowych” spełnia wymagania formalne pracy dyplomowej inżynierskiej.

Gliwice, dnia 9 stycznia 2021

.....
(podpis promotora)

Spis treści

1	Wstęp	1
1.1	Cel pracy	2
1.2	Zakres pracy	2
1.3	Plan pracy	2
2	Analiza dziedziny	5
2.1	Analiza problemu	5
2.2	Konwolucyjne sieci neuronowe	6
2.2.1	Warstwy	6
2.2.2	przykładowe modele	6
2.3	R-CNN	6
2.3.1	Algorytm wyszukiwania selektywnego	6
2.4	ggg	6
3	Wymagania i narzędzia	7
4	Specyfikacja zewnętrzna	9
5	Specyfikacja wewnętrzna	11
6	Weryfikacja i walidacja	13
7	Podsumowanie i wnioski	15

Rozdział 1

Wstęp

W ciągu ostatnich kilku lat można zaobserwować gwałtowny rozwój dziedzin z zakresu uczenia maszynowego oraz sieci neuronowych. Pomimo pozornej nowości tych technologii, podstawy teoretyczne wielu z nich zostały opracowane już w latach 40. zeszłego stulecia [1]. Idee te były sukcesywnie rozwijane oraz modyfikowane, lecz ograniczenia sprzętowe oraz trudność w dostępie do danych uniemożliwiały ich realne wykorzystanie. Dopiero na początku zeszłej dekady postępująca cyfryzacja oraz digitalizacja spowodowała znaczny wzrost ilości przechowywanych danych oraz ich większą dostępność. W tabeli 1.1 pokazano, jak zmieniały się rozmiary wybranych zbiorów danych przeznaczonych do zagadnień związanych z rozpoznawaniem rysów twarzy na przestrzeni lat. Łatwo zauważyć szybko zwiększające się rozmiary kolejnych baz danych, ze szczególnie gwałtownym wzrostem pomiędzy 2008 a 2014 rokiem. Dzięki dostępności coraz to większych zbiorów danych oraz ciągle rosnącej mocy obliczeniowej komputerów, systemy oparte na sztucznej inteligencji osiągają coraz to lepsze wyniki i są w stanie wykonywać pewne zadania lepiej niż człowiek.

W ostatnich latach można zaobserwować zwiększający się wpływ tych systemów na ludzkie życie w wielu różnych dziedzinach, takich jak np. diagnostyce chorób[2], [3], samo-prowadzących się pojazdach, cyberbezpieczeństwie, czy marketingu. Te dotychczasowe osiągnięcia systemów opartych o sztuczną inteligencję oraz potencjał ten dziedziny pozwala przypuszczać, że ich znaczenie w świecie będzie już tylko rosnąć.

Nazwa	Rok powstania	Ilość obrazów
Yale Face Database	1997	165
JAFFE Facial Expression Database	1998	213
Face Recognition Grand Challenge Dataset	2004	4007
CASIA 3D Face Database	2007	4624
Bosphorus	2008	4652
FaceScrub	2014	107818
IMDB-WIKI	2015	523051
Aff-Wild	2017	~ 1,250,000
Aff-Wild2	2019	~ 2,800,000

Tablica 1.1: Rozmiary zbiorów danych służących do rozpoznawania twarzy na przestrzeni lat

1.1 Cel pracy

Celem pracy jest stworzenie uniwersalnego narzędzia, które ma umożliwić ekstrakcję wektorów cech głębokich w postaci serializowanej wraz z przypisanymi do nich etykietami w wybranym przez użytkownika formacie. Ekstrakcja jest dokonywana za pomocą konwolucyjnych sieci neuronowych służących do detekcji obiektów. Narzędzie powinno mieć możliwość wyboru architektury sieci, jak i dodania własnych architektur. Domyślną architekturą systemu, która zostanie zaimplementowana będzie architektura R-CNN. Narzędzie ma mieć możliwość użycia własnego zestawu danych w formacie PASCAL-VOC.

1.2 Zakres pracy

Zakres pracy obejmuje zgłębienie dziedziny wizji komputerowej oraz przegląd literatury technicznej. Kolejnym krokiem jest dogłębne zrozumienie konwolucyjnych sieci neuronowych służących do detekcji obiektów w obrazach, a następnie zapoznanie się bazą danych PASCAL-VOC oraz formatem przechowywanych tam danych. Kolejnym etapem jest przegląd oraz wybór odpowiedniej technologii.

1.3 Plan pracy

Praca składa się z 7 rozdziałów, które opisują teoretyczne oraz praktyczne

ujęcie tematu. Rozdział 1 zawiera wstęp do tematu oraz określenie celów projektu. Rozdział 2 składa się z analizy zagadnienia detekcji obiektów w obrazach, przeglądu i porównanie dotychczas znanych rozwiązań i technologii. W rozdziale 3 omówiono wymagania funkcjonalne i нефункционалне oraz dokonano opisu zastosowanych narzędzi. Rozdział 4 obejmuje specyfikację zewnętrzną. Zostanie w nim opisany sposób instalacji oraz przykładowe scenariusze korzystania z narzędzia. W rozdziale 5 można znaleźć opis architektury systemu oraz omówienie użytych modułów i bibliotek. Rozdział 6 zawiera opis weryfikacji oraz walidacji systemu. W rozdziale 7 zawarto podsumowanie całej pracy oraz wnioski z niej płynące. Wymieniono również największe trudności, które napotkano w czasie pracy nad narzędziem.

Rozdział 2

Analiza dziedziny

W tym rozdziale zostanie omówiony problem detekcji oraz klasyfikacji obiektów w obrazach. Pokróćce wyjaśniona zostanie zasada działania konwolucyjnych sieci neuronowych, ze zwięzłym opisem różnych rodzajów warstw, a następnie przedstawione zostaną kilka najważniejszych modeli sieci neuronowych. Opisana zostanie architektura R-CNN, która została zaimplementowana w programie, oraz algorytm wyszukiwania selektywnego, który również został zaimplementowany w ramach tej architektury. Aby móc uzyskać jakieś porównanie co do wydajności i ograniczeń architektury R-CNN, pokazane zostaną również inne architektury sieci, takie jak Fast R-CNN czy YOLO.

2.1 Analiza problemu

Człowiek postrzega świat głównie wizualnie. Szacuje się, że 80 % bodźców odbieranych przez człowieka to bodźce wzrokowe. Niektóre z teorii [4] pozwalają przypuszczać, że wykształcenie oka było najważniejszym momentem w historii ewolucji oraz kluczowym elementem, który umożliwił powstanie inteligentnych form życia. Nic więc dziwnego, że temat tak znaczący dla człowieka otrzymuje proporcjonalnie dużo uwagi w dziedzinie sztucznej inteligencji. Umożliwienie maszynom zrozumienia wizualnych danych jest głównym celem, do którego spełnienia jesteśmy, zdawałoby się mogło, coraz bliżej. Jednym z podstawowych problemów z dziedziny wizji komputerowej jest klasyfikacja. Polega ona na przypisaniu pewnej kategorii

na podstawie obrazu. Zazwyczaj kategorie te to obiekty znajdujące się na zdjęciu. Chcemy więc, aby maszyna po zobaczeniu zdjęcia psa skategoryzowała go jako 'pies'. Do problemu klasyfikacji możemy dołożyć jeszcze inny problem - detekcji. Teraz chcemy, aby maszyna po zobaczeniu zdjęcia psa nie tylko zidentyfikowała go jako psa, lecz również wskazała w którym miejscu na zdjęciu ten pies się znajduje.

2.2 Konwolucyjne sieci neuronowe

2.2.1 Warstwy

2.2.2 przykładowe modele

AlexNet

ImageNet

VGG16

ResNet

2.3 R-CNN

2.3.1 Algorytm wyszukiwania selektywnego

2.4 ggg

- analiza tematu
- wprowadzenie do dziedziny (*state of the art*) – sformułowanie problemu
- studia literaturowe
- przegląd literatury tematu (należy wskazać źródła wszystkich informacji zawartych w pracy)
- opis znanych rozwiązań (także opisanych naukowo, jeżeli problem jest poruszany w publikacjach naukowych), algorytmów, osadzenie pracy w kontekście

Rozdział 3

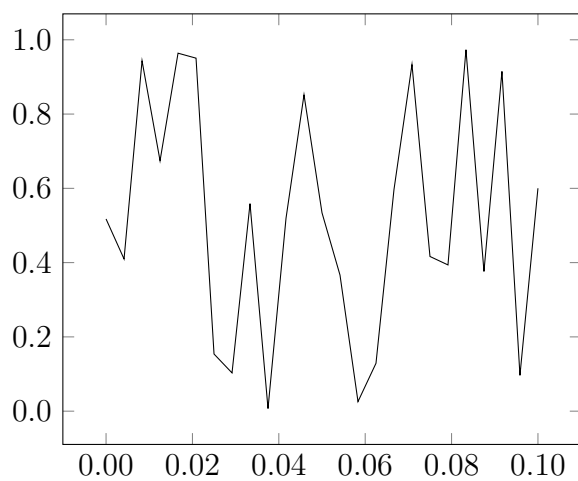
Wymagania i narzędzia

- wymagania funkcjonalne i нефункционалне
- przypadki użycia (diagramy UML)
- opis narzędzi
- metodyka pracy nad projektowaniem i implementacją

Rozdział 4

Specyfikacja zewnętrzna

- wymagania sprzętowe i programowe
- sposób instalacji
- sposób aktywacji
- kategorie użytkowników
- sposób obsługi
- administracja systemem
- kwestie bezpieczeństwa
- przykład działania
- scenariusze korzystania z systemu (ilustrowane zrzutami z ekranu lub generowanymi dokumentami)



Rysunek 4.1: Podpis rysunku po rysunkiem.

Rozdział 5

Specyfikacja wewnętrzna

- przedstawienie idei
- architektura systemu
- opis struktur danych (i organizacji baz danych)
- komponenty, moduły, biblioteki, przegląd ważniejszych klas (jeśli występują)
- przegląd ważniejszych algorytmów (jeśli występują)
- szczegóły implementacji wybranych fragmentów, zastosowane wzorce projektowe
- diagramy UML

Krótką wstawka kodu w linii tekstu jest możliwa, np. **descriptor**, a nawet **descriptor_gaussian**. Dłuższe fragmenty lepiej jest umieszczać jako rysunek, np. kod na rysunku 5.1, a naprawdę długie fragmenty – w załączniku.

```
1 class descriptor_gaussian : virtual public descriptor
2 {
3     protected:
4         /** core of the gaussian fuzzy set */
5         double _mean;
6         /** fuzzyfication of the gaussian fuzzy set */
7         double _stddev;
8
9     public:
10        /** @param mean core of the set
11            @param stddev standard deviation */
12        descriptor_gaussian (double mean, double stddev);
13        descriptor_gaussian (const descriptor_gaussian & w);
14        virtual ~descriptor_gaussian();
15        virtual descriptor * clone () const;
16
17        /** The method elaborates membership to the gaussian
18            fuzzy set. */
19        virtual double getMembership (double x) const;
20    };
```

Rysunek 5.1: Klasa **descriptor_gaussian**.

Rozdział 6

Weryfikacja i walidacja

- sposób testowania w ramach pracy (np. odniesienie do modelu V)
- organizacja eksperymentów
- przypadki testowe zakres testowania (pełny/niepełny)
- wykryte i usunięte błędy
- opcjonalnie wyniki badań eksperymentalnych

Rozdział 7

Podsumowanie i wnioski

- uzyskane wyniki w świetle postawionych celów i zdefiniowanych wyżej wymagań
- kierunki ewentualnych danych prac (rozbudowa funkcjonalna ...)
- problemy napotkane w trakcie pracy

Tablica 7.1: Opis tabeli nad nią.

ζ	metoda						
	alg. 1	alg. 2	alg. 3			alg. 4, $\gamma = 2$	
			$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724

Bibliografia

- [1] W. S. M. W. P. i in., “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, pp. 115–133, 1943.
- [2] S. M. i in., “International evaluation of an ai system for breast cancer screening,” *Nature*, vol. 577, no. 7788, p. 89–94, 2020.
- [3] K. D. i in., “Effect of artificial intelligence-based triaging of breast cancer screening mammograms on cancer detection and radiologist workload: a retrospective simulation study,” *The Lancet*, vol. 2, no. 9, pp. e468–e474, 2020.
- [4] D.-E. Nilsson, “Eye evolution and its functional basis,” *Visual neuroscience*, vol. 30, no. 1-2, pp. 5–20, 2013.

Dodatki

Spis skrótów i symboli

DNA kwas deoksyrybonukleinowy (ang. *deoxyribonucleic acid*)

MVC model – widok – kontroler (ang. *model-view-controller*)

N liczebność zbioru danych

μ stopień przyleżności do zbioru

\mathbb{E} zbiór krawędzi grafu

\mathcal{L} transformata Laplace’a

Źródła

Jeżeli w pracy konieczne jest umieszczenie długich fragmentów kodu źródłowego, należy je przenieść do załącznika.

```
1 partition fcm_possibilistic :: doPartition
2                               (const dataset & ds)
3 {
4     try
5     {
6         if (_nClusters < 1)
7             throw std::string ("unknown_number_of_clusters");
8         if (_nIterations < 1 and _epsilon < 0)
9             throw std::string ("You should set a maximal
10                                number_of_iteration_or_minimal_difference_or_
11                                epsilon.");
12
13         if (_nIterations > 0 and _epsilon > 0)
14             throw std::string ("Both number_of_iterations_and_
15                                minimal_epsilon_set_or_you should set either_
16                                number_of_iterations_or_minimal_epsilon.");
17
18         auto mX = ds.getMatrix();
19         std::size_t nAttr = ds.getNumberOfAttributes();
20         std::size_t nX    = ds.getNumberOfData();
21         std::vector<std::vector<double>> mV;
22         mU = std::vector<std::vector<double>> (_nClusters);
23         for (auto & u : mU)
```



```
19         u = std::vector<double> (nX);
20     randomise(mU);
21     normaliseByColumns(mU);
22     calculateEtas(_nClusters, nX, ds);
23     if (_nIterations > 0)
24     {
25         for (int iter = 0; iter < _nIterations; iter++)
26         {
27             mV = calculateClusterCentres(mU, mX);
28             mU = modifyPartitionMatrix (mV, mX);
29         }
30     }
31     else if (_epsilon > 0)
32     {
33         double frob;
34         do
35         {
36             mV = calculateClusterCentres(mU, mX);
37             auto mUnew = modifyPartitionMatrix (mV, mX);
38
39             frob = Frobenius_norm_of_difference (mU, mUnew)
40                 ;
41             mU = mUnew;
42         } while (frob > _epsilon);
43     }
44     mV = calculateClusterCentres(mU, mX);
45     std::vector<std::vector<double>> mS =
46         calculateClusterFuzzification(mU, mV, mX);
47
48     partition part;
49     for (int c = 0; c < _nClusters; c++)
50     {
51         cluster cl;
```


Zawartość dołączonej płyty

Do pracy dołączona jest płyta CD z następującą zawartością:

- praca (źródła \LaTeX owe i końcowa wersja w pdf),
- źródła programu,
- dane testowe.

Spis rysunków

4.1	Podpis rysunku po rysunkiem.	10
5.1	Klasa descriptor_gaussian	12

Spis tablic

1.1	Rozmiary zbiorów danych służących do rozpoznawania twarzy na przestrzeni lat	2
7.1	Opis tabeli nad nią.	16