

XVI Konferencja Naukowo-Techniczna

TKI2022

TECHNIKI KOMPUTEROWE W INŻYNIERII

18-21 października 2022

Detekcja ludzi przy pomocy kamery termowizyjnej i głębokich sieci neuronowych

Artur Stefańczyk¹, Julia Zięba¹

¹Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Inżynierii Mechanicznej i Robotyki
email: astefanczyk@student.agh.edu.pl, juliazieba@student.agh.edu.pl

STRESZCZENIE:

Na przestrzeni ostatnich lat sztuczne sieci neuronowe odnalazły wiele zastosowań, między innymi w analizie danych wizyjnych, np. możliwość detekcji ludzi. Najczęściej w tym celu wykorzystuje się obraz z kamer RGB, jednakże w niektórych przypadkach konieczne jest zastosowanie innych czujników wizyjnych. W tym artykule przedstawimy wyniki analizy danych wizyjnych z kamery termowizyjnej przy wykorzystaniu głębokich sieci neuronowych i warstw konwolucyjnych. Zastosowanie kamery termowizyjnej pozwala na wykrywanie ludzi w niepewnym lub nieregularnym terenie, osiągając większą precyzję niż inne rozwiązania wizyjne. To rozwiązanie może pomóc w wykrywaniu ludzi w górzystym lub zalesionym terenie, przy pomocy dronów lub robotów mobilnych.

SŁOWA KLUCZOWE: sztuczne sieci neuronowe, analiza obrazu, kamera termowizyjna

1. Cele pracy

Wraz z postępem technologicznym powstała potrzeba lepszego zarządzania i analizy danych. Odpowiedzią na ten problem okazały się między innymi sztuczne sieci neuronowe [1]. Inspiracją do ich skonstruowania były ludzkie komórki nerwowe. Pozwalają one na rozwiązywanie wielowymiarowych, nieliniowych problemów optymalizacyjnych [2]. Dzięki tym cechom znalazły szerokie zastosowanie w inżynierii, między innymi w analizie danych wizyjnych [3]. W naszym projekcie umożliwiły one przeprowadzenie testów działania algorytmów klasyfikacji obiektów [4] przechwyconych za pomocą kamery termowizyjnej. Zadanie jest dość złożone ze względu na to, że czas wykonania śledzenia jest krótki, a warunki pogodowe nie zawsze są korzystne. Aby zapobiec błędom stosuje się algorytmy pracujące w trybie real-time posiadające ograniczenie czasowe w jakim muszą wydać wynik. Zastosowane w naszym projekcie rozwiązanie jest często wykorzystywane przez służby militarne, systemy ochrony oraz monitorowanie różnych procesów przemysłowych [5].

2. Metodologia badań

Detekcję ludzi sprawdzić można do zbudowania modelu klasyfikatora binarnego, który na podstawie danych treningowych jest w stanie zadaną skuteczność określić, czy na danym obrazie znajdują się sylwetka człowieka. Uczenie modelu na surowych danych wizyjnych jest często utrudnione, a nawet niemożliwe w związku z czym konieczna jest ekstrakcja cech na podstawie, których przeprowadza się uczenie modelu. Najszerzej stosowanym sposobem ekstrakcji cech obecnie jest stosowanie konwolucyjnych sieci neuronowych (CNN). Ponieważ sieci CNN przetwarzają obraz tylko lokalnie, konieczne jest także ówczesne globalne przetworzenie obrazu, czyli zmiana takich parametrów jak jasność lub kontrast obrazu.

$$c[m, n] = (a * h)[m, n] = \sum_j \sum_k h[j, k] a[m - j, n - k] \quad (1)$$

Konwolucyjne sieci neuronowe są obecnie najskuteczniejszą metodą ekstrakcji cech z obrazu. Metoda ta wykorzystuje operację splotu (1), aby przekształcić obraz w mapę cech. Splot wykonywany jest pomiędzy macierzą reprezentującą monochromatyczny obraz wejściowy, a jądrem (ang. kernel). Jądro w tym przypadku także jest macierzą, o zadanych wymiarach, której elementy zmieniają się wraz z uczeniem modelu. Każdy splot macierzy reprezentującej obraz z jądrem interpretować można jako filtrację obrazu, po której tworzymy nową mapę cech. Mapa cech, z definicji splotu, jest identycznych rozmiarów co obraz wejściowy, gdyż stosowany jest padding [6] pozwalający na wykonanie konwolucji na brzegach obrazu. Wartości elementów mapy cech reprezentują istotność danej cechy.

Ze względu na wysoką wymiarowość problemu bezpośrednie uczenie modelu na mapie cech nie jest zalecane. W celu rozwiązania tego problemu stosuje się algorytm kompresji danych MaxPooling [7] polegający na zmniejszeniu wymiarowości problemu w taki sposób, aby jej najbardziej znaczące elementy zostały zachowane, a także relację pomiędzy nimi.

3. Modelowanie numeryczne

Zbiór danych podzielony został na zbiór treningowy i testowy w proporcji 7:3. Wykorzystano open-source'owe dane, używając 26,442 ujęć z 15 różnymi etykietami. Przykłady przedstawiono na rys. 1. Zastosowano do nich odpowiednią jasność i kontrast. Aby zachować anonimowość i bezpieczeństwo badanych jednostek użyto czujniki widma widzialnego i termicznego („RGBT”). Do uczenia modelu wykorzystano komputer wyposażony w kartę graficzną GeForce RTX3060.

Obrazy ze zbioru treningowego przeskalowano do rozdzielczości 512x512. Sieć konwolucyjna składa się z 3 warstw splotowych pomiędzy którymi stosowany jest algorytm MaxPooling. Następnie Uzyskana mapa cech zostaje poddana operacji flattening, zamian macierzy w wektor. Otrzymany wektor przechodzi przez dwie warstwy wielowarstwowego perceptronu, pomiędzy którego warstwami użyto funkcję aktywacji ReLu [10], gdyż stosowanie proporcjonalnych funkcji aktywacji znacząco ogranicza zagrożenie związane ze znikającym gradientem. Na warstwie wyjściowej znajduje się jeden neuron; jako funkcję aktywacji został zastosowany sigmoid.

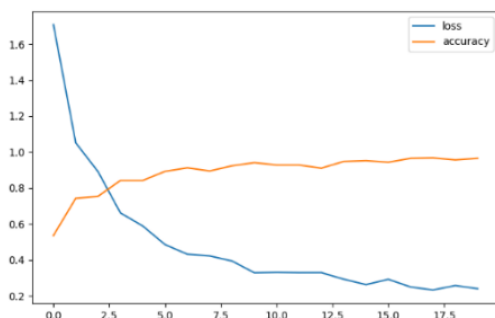


Rys. 1. Dane testowe i treningowe

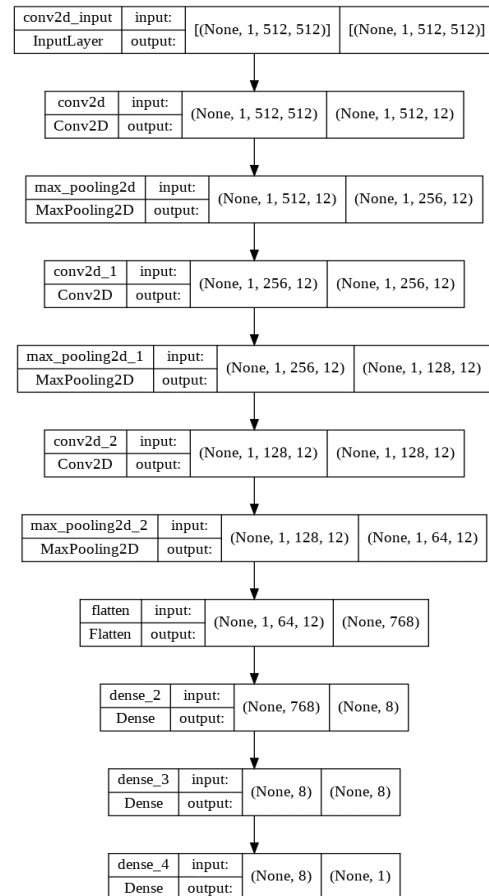
4. Wyniki i analiza

Trening sieci został przeprowadzony przy pomocy biblioteki Tensorflow. Warstwy konwolucyjne modelu posiadają jądro będące macierzą 4x4. Pomiędzy każdą warstwą konwolucyjną zastosowano algorytm MaxPooling w celu wykonania operacji próbkowania przestrzennego i tym samym zmniejszenia wymiarowości problemu, w tym celu użyto macierzy 2x2.

Aby zapobiec przeuczeniu sieci, warstwy konwolucyjne zostały poddane restrukturyzacji L1, o kroku 10^{-3} . Jako optymalizator został zastosowany algorytm Adam o kroku uczenia równym 10^{-3} . Jako funkcję straty zastosowano binarną entropię krzyżową. Pomiędzy warstwami ukrytymi zastosowano funkcję aktywacji ReLu, natomiast na warstwie wyjściowej użyto funkcji aktywacji Sigmoid, w celu zwiększenia dokładności klasyfikatora binarnego. Trening został przeprowadzony w trakcie 20 epok. Dokładność na zbiorze treningowym wyniosła 98,1%. Natomiast dokładność na zbiorze testowym wyniosła 96,7%. Krzywą konwergencji można zobaczyć na rys. 2, a architekturę wykorzystanej sieci neuronowej na rys. 3.



Rys. 2. Krzywa konwergencji z treningu sieci



Rys. 3. Architektura wykorzystanej sieci neuronowej

Praca została wykonana w ramach Rektorskiego nr 71/GRANT/2022 (15/IMIR/2022) Ten projekt został wykonany przez studentów KN NewTech działającego na Wydziale Inżynierii Mechanicznej i Robotyki, w Akademii Górniczo-Hutniczej w Krakowie.

Literatura

- [1] Hongming Wang, Ryszard Czerminski, and Andrew C Jamieson. Neural networks and deep learning. In The machine age of customer insight. Emerald Publishing Limited, 2021.
- [2] Michelle Torres and Francisco Cant u. Learning to See: Convolutional Neural Networks for the Analysis of Social Science Data. Political Analysis, 30(1):113–131, 2022.
- [3] Grace W. Lindsay. Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future. Journal of Cognitive Neuroscience, 33(10):2017–2031, 09 2021.
- [4] People detection and finding attractive areas by the use of movement detection analysis and deep learning approach. Procedia Computer Science, 156:327–337, 2019. 8th International Young Scientists Conference on Computational Science, YSC2019, 24–28 June 2019, Heraklion, Greece.
- [5] Andres Gomez, Francesco Conti, and Luca Benini. Thermal image-based cnn's for ultra-low power people recognition. pages 326–331, 05 2018.
- [6] Nguyen Tuan Nam and Phan Duy Hung. Padding methods in convolutional sequence model: An application in japanese handwriting recognition. ICMLSC 2019, page 138–142, New York, NY, USA, 2019. Association for Computing Machinery.
- [7] F. M. J. Mehedi Shamrat, M. A. Jubair, M. M. Billah, S. Chakraborty, M. Alauddin and R. Ranjan, "A Deep Learning Approach for Face Detection using Max Pooling," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021, pp. 760-764, <https://adas-dataset-v2.flirconservator.com/#downloadguide> https://github.com/LeweLotki/IR_detection