

ZASTOSOWANIE SIECI RESNET DO WYBORU PODZIAŁU JEDNOSTKI CTU W HEVC ALL INTRA

RESNET NETWORK FOR THE CTU UNIT PARTITIONING ALGORITHM IN ALL INTRA HEVC

Jakub Kwaśniak¹; Mateusz Majtko²; Mateusz Lorkiewicz³; Tomasz Grajek⁴

¹ Politechnika Poznańska, Poznań, jakub.kwasniak@student.put.poznan.pl

² Politechnika Poznańska, Poznań, mateusz.majtko@student.put.poznan.pl

³ Politechnika Poznańska, Poznań, mateusz.lorkiewicz@put.poznan.pl

⁴ Politechnika Poznańska, Poznań, tomasz.grajek@put.poznan.pl

Streszczenie: Jednym z zastosowań sztucznych sieci neuronowych jest przyspieszanie istniejących koderów wizyjnych. W niniejszej pracy pokazano zastosowanie zaawansowanej architektury typu ResNet do przyspieszenia algorytmu wyboru podziału jednostek CTU w koderze HEVC pracującym w trybie All Intra. Przedstawione eksperymenty pokazują opłacalność zaproponowanych architektur przy uwzględnieniu sprawności kompresji oraz redukcji czasu kodowania.

Abstract: One application of artificial neural networks is to accelerate existing video encoders. This paper demonstrates the use of ResNet type architectures to accelerate the CTU unit partitioning algorithm in HEVC in All Intra mode. The experiments presented show the cost-effectiveness of the proposed architectures considering compression efficiency and coding time reduction.

Słowa kluczowe: HEVC, kodowanie wizji, ResNet, sztuczne sieci neuronowe.

Keywords: artificial neural networks, HEVC, ResNet, video encoding.

1. WSTĘP

Sygnał wizyjny – obrazy zarówno statyczne, jak i ruchome – stanowi coraz większy udział w globalnym ruchu internetowym [17]. Do głównych źródeł tego ruchu należą telewizja, wideo na żądanie (Video on Demand), media społecznościowe oraz platformy streamingowe. Zwiększająca się liczba źródeł treści wizyjnych wiąże się ze wzrostem liczby urządzeń odpowiedzialnych za ich kompresję. Znaczną część z nich stanowią urządzenia mobilne, charakteryzujące się ograniczonymi zasobami obliczeniowymi i energetycznymi. W rezultacie rośnie zapotrzebowanie na nowoczesne, energooszczędne metody kompresji sygnału wizyjnego.

Jedną z najnowszych technik stosowanych w tym celu jest High Efficiency Video Coding (HEVC) [10], wykorzystywana m.in. w telewizji cyfrowej. Pomimo przyjęcia normy w 2013 roku, nadal rozwijane są nowe algorytmy sterujące pracą koderów, umożliwiające znaczące przyspieszenie procesu kompresji.

Wysoka efektywność kompresji w HEVC wynika m.in. z hierarchicznego podziału obrazu na mniejsze bloki. W pierwszym etapie obraz dzielony jest na bloki o stałej wielkości, tzw. Coding Tree Units (CTU), które następnie są dzielone na mniejsze jednostki – Coding Units (CU), poddawane właściwej kompresji. Proces ten, określany jako partycjonowanie, polega na wyszukiwaniu

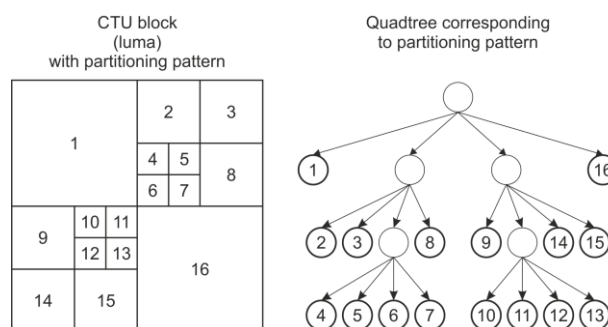
optymalnego podziału bloku CTU na jednostki CU. Jest to najbardziej czasochłonny etap kodowania [4].

W literaturze opisano liczne metody mające na celu przyspieszenie procesu partycjonowania, spośród których najskuteczniejsze wykorzystują sztuczne sieci neuronowe. Dotychczas stosowano głównie proste architektury, takie jak AlexNet [13] czy w pełni splotowe sieci neuronowe [7]. W ostatnich latach zaproponowano jednak bardziej zaawansowane modele, takie jak ResNet [8], które osiągają lepsze wyniki w zadaniach uczenia maszynowego.

W niniejszej pracy przedstawiono zastosowanie architektury ResNet w algorytmie partycjonowania kodera HEVC działającego w trybie All Intra.

2. PRZEGLĄD LITERATURY

Jednym z kluczowych elementów techniki HEVC [10] jest podział bloków CTU na mniejsze jednostki kodowania. Blok CTU najczęściej ma rozmiar 64×64 próbek składowej luma. Podstawową jednostką kodowania jest Coding Unit (CU), która określa sposób kompresji próbek w obrębie bloku. Podział CTU realizowany jest zgodnie z drzewem czwórkowym (quadtree). Przykładowy schemat podziału bloku CTU na jednostki CU przedstawiono na Rysunku 1.



Rys. 1. Przykładowy podział bloku CTU na jednostki CU z odpowiadającym drzewem czwórkowym

Informacja o podziale bloku CTU prezentowana jest najczęściej na dwa sposoby. W strumieniu bitowym sposób podziału sygnalizowany jest za pomocą flag podziału. W implementacjach koderów lub dekoderów często używa się reprezentacji przy pomocy macierzy podziału – kwadratowej macierzy o rozmiarze 16×16 (dla CTU

64×64). Wartości w macierzy podziału określają głębokość jednostki CU w drzewie czwórkowym.

W literaturze zaproponowano wiele algorytmów szybkiego wyznaczania podziału CTU na jednostki CU. Celem tych algorytmów jest znaczące skrócenie czasu kodowania przy możliwie niewielkim pogorszeniu efektywności kompresji. Kompromis ten ocenia się najczęściej poprzez porównanie z koderem referencyjnym HEVC, oznaczanym akronimem HM [9].

Wśród istniejących podejść znaleźć można algorytmy, które podejmują decyzje o podziale na podstawie różnych cech obrazu, takich jak tekstura [18], cechy statystyczne [16], czy cechy heurystyczne [12]. Rozwiązania te najczęściej służą do rekursywnego wyznaczania flag podziału.

Wraz z rozwojem metod uczenia maszynowego pojawiły się algorytmy oparte na sztucznych sieciach neuronowych. Najczęściej stosowane są proste architektury splotowe, takie jak AlexNet, wykorzystywane do predykcji flag podziału [2, 6]. Zaproponowano również ich modyfikacje umożliwiające jednocześnie wyznaczanie wszystkich flag dla całego bloku CTU [5] lub bardziej zaawansowane architektury, takie jak ResNet [14].

Wraz z popularyzacją sieci w pełni splotowych, zaproponowano modele umożliwiające bezpośrednie wyznaczanie całej macierzy podziału [15]. Algorytmy tego typu, oparte na sieciach splotowych, należą obecnie do najskuteczniejszych metod szybkiego podziału bloków CTU, oferując korzystny kompromis pomiędzy czasem kodowania a efektywnością kompresji. Zazwyczaj opierają się one na prostych architekturach sieci w pełni splotowych.

3. PROPONOWANE ROZWIĄZANIE

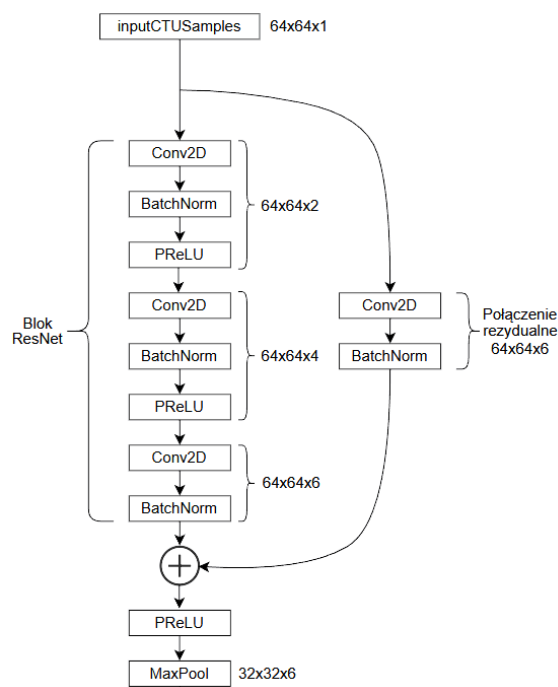
Niniejsza praca koncentruje się na algorytmach szybkiego podziału bloków CTU, których celem jest bezpośrednie wyznaczanie całej macierzy podziału w koderze HEVC działającym w trybie All Intra. Głównym zagadnieniem badawczym jest wykorzystanie sieci neuronowej typu ResNet w procesie predykcji struktury podziału bloku CTU.

W badaniach wykorzystano zmodyfikowaną wersję kodera HEVC, udostępnioną jako część publikacji [15]. W związku z tym architektura wejścia i wyjścia projektowanych sieci została dostosowana do wymagań zgodnych z tym oprogramowaniem.

Sieć typu ResNet [8] jest szczególnym przykładem sieci w pełni splotowej. Jej charakterystyczną cechą jest zastosowanie połączeń rezydualnych pomiędzy wejściem a wyjściem bloku warstw splotowych (kilka połączonych ze sobą warstw splotowych). Dzięki takiemu połączeniu do kolejnego bloku przekazywana jest mapa cech będąca sumą wejścia i wyjścia z danego bloku warstw splotowych. Architektura ta umożliwia skuteczniejsze i szybsze uczenie modelu, często pozwalając na osiągnięcie lepszych rezultatów niż klasyczne sieci w pełni splotowe.

Zaproponowane w niniejszej pracy sieci neuronowe oparto na architekturze ResNet, składającej się z kilku

bloków rezydualnych (Rys. 2). Na wyjściu każdej sieci zastosowano funkcję aktywacji SoftMax. Szczegółowe parametry użytych architektur przedstawiono w Tabeli 1.



Rys. 2. Blok rezydualny wykorzystywany do budowy proponowanych sieci typu ResNet

Danymi wejściowymi dla sieci są próbki luminancji (luma) bloku CTU o rozmiarze 64×64. Wyjście stanowi tensor o wymiarach 4×4×4, reprezentujący prawdopodobieństwa przypisania jednej z czterech możliwych głębokości podziału do każdego z 16 podobszarów CTU. Wymiar 4×4 odpowiada zredukowanej wersji standardowej macierzy podziału (16×16), wynikającej z minimalnego rozmiaru jednostki CU (8×8). Trzeci wymiar tensora wynika z maksymalnej głębokości drzewa czwórkowego, która dla CTU 64×64 wynosi 4.

Aby wykorzystać wyjście sieci w koderze HEVC, konieczna jest konwersja tensora do macierzy o rozmiarze 16×16. Proces ten obejmuje trzy etapy:

- **Wyznaczenie głębokości podziałów**
Zastosowanie funkcji ArgMax wzdłuż trzeciego wymiaru tensora pozwala uzyskać macierz 4×4 zawierającą estymowane głębokości podziałów.
- **Weryfikacja poprawności podziału**
Ponieważ tylko część macierzy 4×4 odpowiada dopuszczalnym strukturom kodera HEVC, stosowana jest heurystyczna procedura korekcyjna. Jeśli liczba komórek o głębokości 0 przekracza 8, cała macierz ustawiana jest na 0. Następnie analizowane są mniejsze regiony: jeśli suma wartości w danym obszarze przekracza określony próg, jego głębokość ustalana jest na 1; w przeciwnym razie – na 2 (z pominięciem wartości 3 jako niezgodnych).
- **Odwzorowanie do formatu 16×16**
Skorygowana macierz 4×4 jest skalowana przestrzennie do formatu 16×16 przez zwielokrotnienie jej wartości w odpowiednich regionach zgodnie ze składnią kodera HEVC.

Tab. 1. Testowane konfiguracje sieci ResNet

ResNet	Opis modyfikacji sieci.	Liczba bloków	Liczba warstw w bloku
Mod 0	Bazowa architektura sieci.	7	8
Mod 1	Zastąpienie funkcji MaxPool2D funkcją AvgPool2D.	7	8
Mod 2	Podwójny blok ResNet przed warstwą aktywacji PReLU oraz warstwą MaxPool.	7	8
Mod 3	Zmniejszenie liczby warstw w bloku rezydualnym do pięciu.	7	5
Mod 4	Zwiększenie dwukrotnie liczby warstw w każdym bloku.	7	14
Mod 5	Zwiększenie o cztery warstwy w każdym bloku.	7	11
Mod 6	Potrójny blok ResNet przed warstwą aktywacji PReLU oraz warstwą MaxPool.	7	8
Mod 7	Wykorzystanie ReLU jako funkcji aktywacji.	7	8
Mod 8	Zastosowanie ośmiu bloków rezydualnych.	8	8
Mod 9	Zastosowanie sześciu zamiast rezydualnych.	6	8
Mod 10	Szybsze zwiększanie liczby wyjściowych map cech w kolejnych blokach rezydualnych.	7	8

4. METODOLOGIA BADAŃ

Do trenowania zaproponowanych architektur wykorzystano zbiór DIV2k [1]. Obrazy zostały zakodowane za pomocą referencyjnego kodera HM [9], a następnie podzielone na bloki CTU z przypisanymi podziałami wyznaczonymi przez HM. Zbiór obejmuje 589 589 przykładów treningowych oraz 66 650 przykładów walidacyjnych. Dla każdej wartości parametru kwantyzacji $QP = \{22, 27, 32, 37\}$, zgodnie z Common Test Conditions (CTC) dla HEVC [11], przygotowano osobny podzbiór danych uczących.

Sieci trenowano w środowisku TensorFlow, wykorzystując entropię krzyżową jako funkcję straty oraz optymalizator ADAM. Model uczono przez 50 epok z resetowaniem optymalizatora co 5 epok, korzystając z akceleracji GPU.

Ewaluację przeprowadzono za pomocą zmodyfikowanego kodera HEVC [15] po uprzedniej konwersji modeli do formatu kompatybilnego z LibTorch. Obliczenia podczas testów wykonywano wyłącznie na CPU. Do oceny wykorzystano zbiór danych testowych JCT-VC opisany w CTC, stosując standardową procedurę ewaluacji HEVC [11].

Wydajność sieci oceniono na podstawie: liczby operacji dodawania i mnożenia (w milionach [M]), liczby parametrów (w tysiącach [k]), zmiany strumienia bitowego względem kodera referencyjnego (delta Bjøntegaarda [3]) oraz redukcji czasu kodowania względem kodera referencyjnego (w procentach).

5. WYNIKI EKSPERYMENTALNE

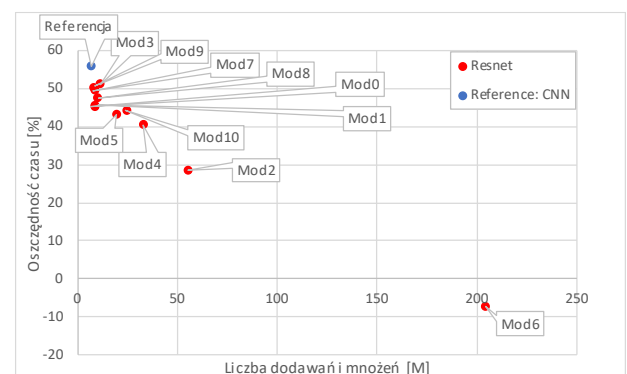
W Tabeli 2 przedstawiono wyniki eksperymentalne dla zaproponowanych architektur sieci typu ResNet. Dla porównania przedstawiono wyniki i parametry dla jednej z najlepszych sieci w pełni splotowej dostępnych w literaturze [15]. Na Rysunkach 3-5 przedstawiono wyniki w formie graficznej, zestawiając ze sobą wybrane rezultaty.

Na podstawie Rysunku 3 obserwuje się, że zaproponowane sieci ResNet, mimo porównywalnej liczby dodawań i mnożeń (np. Mod9, Mod3, Mod7), wykazują mniejszą redukcję czasu kodowania niż rozwiązanie refe-

rencyjne. Wyjątkiem jest Mod6, którego złożoność powoduje wydłużenie czasu kodowania względem kodera referencyjnego. Podobne wnioski płyną z analizy oszczędności czasu kodowania (Rysunek 4). Przyczyną może być większa liczba warstw w sieciach ResNet, co skutkuje większą serializacją obliczeń i wolniejszym przetwarzaniem, mimo zbliżonej liczby operacji i parametrów.

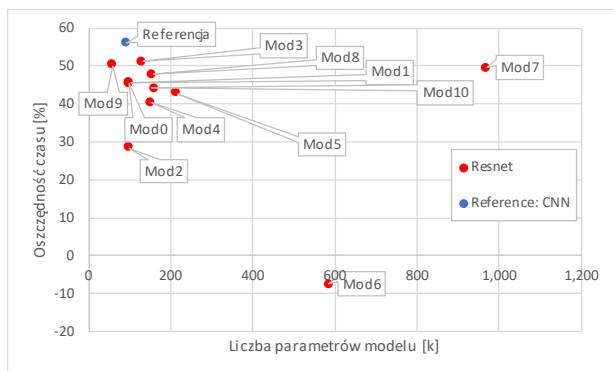
Tab. 2. Wyniki eksperymentalne

ResNet	Liczba dodawań i mnożeń	Liczba parametrów modelu	BD-Rate	Oszczędność czasu kodowania
	[M]	[k]	[%]	[%]
Mod0	8,74	97,202	1,88	45,41
Mod1	8,74	97,202	1,81	45,74
Mod2	55,00	97,148	1,70	28,54
Mod3	11,13	128,188	1,87	51,27
Mod4	32,66	150,992	1,77	40,47
Mod5	19,44	212,878	1,81	43,22
Mod6	204,08	583,968	1,71	-7,46
Mod7	8,74	96,852	1,91	49,59
Mod8	9,62	152,960	1,88	47,67
Mod9	8,10	57,086	1,80	50,36
Mod10	24,49	160,262	1,77	44,28
[15]	6,76	91,600	1,87	56,08

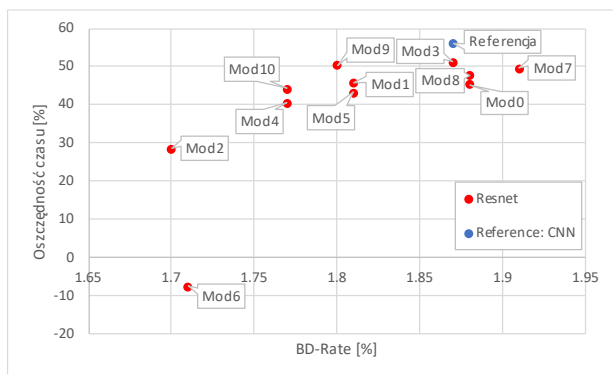


Rys. 3. Oszczędność czasu kodowania w zależności od liczby dodawań i mnożeń dla proponowanych sieci

Jednakże sieci ResNet przewyższają rozwiązanie referencyjne pod względem zmniejszenia wielkości strumienia bitowego (Rysunek 5). Dla większości modeli strumień jest mniejszy o około 0,1 p.p. względem sieci CNN z [15]. Model Mod9 osiąga zbliżoną oszczędność czasu kodowania przy wyraźnie lepszej kompresji w porównaniu do sieci referencyjnej, co potwierdza wyższą efektywność ResNet w zadaniach klasyfikacji.



Rys. 4. Wykres oszczędności czasu kodowania w zależności od liczby parametrów dla proponowanych sieci



Rys. 5. Wykres oszczędności czasu kodowania w zależności od zmiany wielkości strumienia bitowego dla proponowanych sieci.

6. WNIOSKI

Niniejsza praca przedstawia wykorzystanie sztucznych sieci neuronowych typu ResNet jako algorytmu wyznaczania podziału dla bloku CTU w koderze HEVC w trybie All Intra. W pracy przedstawiono wyniki ewaluacji dla kilku zaproponowanych wariantów sieci typu ResNet.

W porównaniu do referencyjnego rozwiązania z zastosowaniem prostszej, w pełni splotowej architektury, zaproponowane sieci typu ResNet okazały się mniej wydajne pod względem oferowanej redukcji czasu kodowania. Takie wyniki otrzymano pomimo zastosowania sieci o porównywalnej liczbie parametrów i ilości obliczeń.

W artykule pokazano, że pomimo słabszej redukcji czasu kodowania, zastosowanie sieci typu ResNet pozwala na osiągnięcie wyraźnie lepszych wyników kodowania pod względem wielkości strumienia bitowego, w porównaniu do referencyjnej sieci w pełni splotowej. Wyniki te wskazują na dalszą konieczność prowadzenia badań nad wykorzystaniem bardziej zaawansowanych architektur sieci neuronowych w opisywanym zastosowaniu.

LITERATURA

[1] Agustsson E., Timofte R.. 2017. „NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study”. *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*. 1122-1131.

[2] Amna M., Imen W., Ezahra S. F.. 2021. „LeNet5-Based approach for fast intra coding”. *10th Int. Symp. on Signal, Image, Video and Communications*. 1-4.

[3] Bjøntegaard G.. 2001. „Calculation of average PSNR differences between RD curves”, *ITU-T SG16/Q6, dokument. VCEG-M33*.

[4] Bossen F., Bross B., Suhring K., Flynn D., 2012. „HEVC Complexity and Implementation Analysis”. *IEEE Transactions on Circuits and Systems for Video Technology*. 22 (12) : 1685-1696.

[5] Chen Z., Shi J., Li W.. 2020 „Learned Fast HEVC Intra Coding”. *IEEE Transactions on Image Processing*. 29 : 5431-5446.

[6] Feng Z., Liu P., Jia K., Duan K.. 2018. „Fast Intra CTU Depth Decision for HEVC”. *IEEE Access*. 6 : 45262-45269.

[7] Goodfellow I., Bengio Y., Courville Aaron. 2016. „Deep Learning”. *MIT Press*.

[8] He K., Zhang X., Ren S., Sun J.. 2016. „Deep Residual Learning for Image Recognition”. *IEEE Conf. on Computer Vision and Pattern Recognition*. 770-778.

[9] HHI. „HEVC reference software”. <https://hevc.hhi.fraunhofer.de>.

[10] ISO/IEC. 2013. „Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High Efficiency Video Coding”. *ISO/IEC IS 23008-2, także jako ITU-T Rec. H.265*.

[11] Joint Collaborative Team Video Coding (JCT-VC). 2013. „Common Test Conditions and Software Reference Configurations”, *dokument JCTVC-L1100*.

[12] Kim N., Jeon S., Shim H. J., Jeon B., Lim S.-Ch., Ko H.. 2016. „Adaptive keypoint-based CU depth decision for HEVC intra coding”. *IEEE Int. Symp. Broadband Multimedia Syst. Broadcast*. 1-3.

[13] Krizhevsky A., Sutskever I., Hinton G. E. 2012. „ImageNet classification with deep convolutional neural networks”, *25th Int. Conf. on Neural Information Processing Systems*, 1097-1105.

[14] Li Y., Li L., Zhuang Z., Fang Y., Yang Y.. 2021. „ResNet Approach for Coding Unit Fast Splitting Decision of HEVC Intra Coding”. *IEEE Sixth Int. Conf. on Data Science in Cyberspace*. 130-135.

[15] Lorkiewicz M., Stankiewicz O., Domański M., Hang H.-M., Peng W.-H.. 2024. „Complexity-Efficiency Control With ANN-Based CTU Partitioning for Video Encoding”. *IEEE Access*. 12 : 102536-102551.

[16] Lu Y., Huang X., Liu H., Zhou Y., Yin H., Shen L.. 2020. „Hierarchical classification for complexity reduction in HEVC inter coding”. *IEEE Access*. 8 : 41690-41704.

[17] Sandvine. 2023. „Sandvine's 2023 Global Internet Phenomena Report Shows 24% Jump in Video Traffic, with Netflix Volume Overtaking YouTube”.

[18] Zhang Q., Wang Y., Huang L., Jiang B.. 2020. „Fast CU partition and intra mode decision method for H.266/VVC”. *IEEE Access*. 8 : 117539-117550.