



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

KOMPRESSE 360° OBRAZU ZALOŽENÁ NA HLUBOKÉM UČENÍ

DEEP LEARNING BASED COMPRESSION OF 360° IMAGES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Adam Budáč

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Kufa, Ph.D.

BRNO 2023

Diplomová práce

magisterský navazující studijní program **Elektronika a komunikační technologie**

Ústav radioelektroniky

Student: Bc. Adam Budáč

ID: 211308

Ročník: 2

Akademický rok: 2022/23

NÁZEV TÉMATU:

Kompresie 360° obrazu založená na hlubokém učení

POKYNY PRO VYPRACOVÁNÍ:

V teoretické části práce podejte přehled o kompresi 360° obrazu založeného na využití metod strojového, případně hlubokého učení. Prostudujte, jaké jsou dominantní deformace u 360° záběrů. Vytvořte framework, pomocí kterého bude možné komprimovat snímky, počítat objektivní metriky, a to plně automatizovaně. Pro Vámi vytvořený testovací data-set vyberte vhodné objektivní a subjektivní hodnotící metriky kvality 360°obrazů. Použitelnost vybraných metod experimentálně vyzkoušejte.

V experimentální části práce, pomocí vytvořených postupů, zrealizujte kompresi 360° obrazů za pomoci hlubokého učení. Pomocí vhodných objektivních metrik, případně subjektivních testů, porovnejte účinnost vytvořených metod a konvenčních algoritmů. Výstupy práce přehledně zpracujte a detailně diskutujte.

DOPORUČENÁ LITERATURA:

[1] XU, Mai, Chen LI, Shanyi ZHANG a Patrick Le CALLET. State-of-the-Art in 360° Video/Image Processing: Perception, Assessment and Compression. IEEE Journal of Selected Topics in Signal Processing [online]. 2020, 14(1), 5-26 [cit. 2022-05-19]. ISSN 1932-4553. Dostupné z: doi:10.1109/JSTSP.2020.2966864

[2] SIMKA, Marek, Jan KUFA a Ladislav POLAK. Picture Quality of 360° Images Compressed by Emerging Compression Algorithms. In: 2022 32nd International Conference Radioelektronika (RADIOELEKTRONIKA) [online]. IEEE, 2022, 2022-4-21, s. 1-4 [cit. 2022-05-19]. ISBN 978-1-7281-8686-3. Dostupné z: doi:10.1109/RADIOELEKTRONIKA54537.2022.9764941

Termín zadání: 6.2.2023

Termín odevzdání: 22.5.2023

Vedoucí práce: Ing. Jan Kufa, Ph.D.

doc. Ing. Lucie Hudcová, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Táto diplomová práca sa zaoberá kompresiou 360° obrazu založenou na hlbokom učení. V práci je popísaná reprezentácia a deformácie 360° obrazu. Ďalej sú popísané konvenčné kompresné metódy a metódy využívajúce strojové a hlboké učenie a hlboké neurónové siete na kompresiu 360° obrazu. Súčasťou práce je vytvorenie datasetu desiatich 360° obrázkov a návrh frameworku umožňujúceho kompresiu obrázkov konvenčnými kodekmi a algoritmami založenými na strojovom a hlbokom učení a hlbokých neurónových sieťach. Obrázky z datasetu sú skomprimované piatimi konvenčnými kodekmi a štyrmi algoritmami hlbokého učenia. Kvalita komprimovaných obrázkov je zameraná siedmymi objektívnymi metrikami a subjektívnym testom. Výsledkom experimentu je, že konvenčné metódy dosiahli vyššiu kvalitu pri kompresii, ako metódy využívajúce hlboké učenie a výpočtová náročnosť konvenčných metód je nižšia v porovnaní s metódami využívajúce hlboké učenie.

Kľúčové slová

Kompresia 360° obrazu, dataset, strojové učenie, hlboké učenie, hodnotenie kvality obrazu, objektívne a subjektívne metriky kvality, framework, Python

Abstract

This diploma thesis deals with 360° image compression based on deep learning. The thesis describes the representation and deformations of a 360° image. It then describes conventional compression methods and methods using machine learning, deep learning and deep neural networks for 360° image compression. Part of the work is the creation of a dataset, which consists of ten 360° images, and the design of a framework that enables image compression using conventional codecs and algorithms based on machine learning, deep learning, and deep neural networks. Images from the dataset are compressed using five conventional codecs and four deep learning algorithms. The quality of compressed images is measured using seven objective metrics and one subjective test. As a result of the experiment, the conventional methods achieved higher compression quality than the methods using deep learning, and the computational complexity of the conventional methods is lower compared to the methods using deep learning.

Keywords

360° image compression, dataset, machine learning, deep learning, image quality assessment, objective and subjective quality metrics, framework, Python

Bibliografická citácia

BUDÁČ, Adam. *Kompresa 360° obrazu založená na hlubokém učení* [online]. Brno, 2023 [cit. 2023-02-06]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/144264>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Vedoucí práce Jan Kufa.

Prehlásenie autora o pôvodnosti diela

Meno a priezvisko študenta: *Adam Budáč*

VUT ID študenta: *211308*

Typ práce: *Diplomová práca*

Akademický rok: *2022/23*

Téma záverečnej práce: *Kompresa 360° obrazu založená na
hlubokém učení*

Prehlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúcej/ho záverečnej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovení § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení časti druhej, hlavy VI. diel 4 Trestného zákoníku č. 40/2009 Sb.

V Brne dňa: 22. mája 2023

podpis autora

Pod'akovanie

Ďakujem vedúcemu diplomovej práce Ing. Jánovi Kufovi, Ph.D. za účinnú metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojej diplomovej práce.

V Brne dňa: 22. mája 2023

podpis autora

Obsah

ZOZNAM OBRÁZKOV	10
ZOZNAM TABULIEK	11
ÚVOD	12
1. REPREZENTÁCIA 360° OBRAZU.....	13
1.1 VYUŽITIE 360° OBRAZU	13
1.2 CHARAKTERISTIKY 360° OBRAZU	14
1.3 OBLASŤ ZÁUJMU V 360° OBRAZE	14
1.4 DEFORMÁCIE 360° OBRAZU	14
1.5 PROJEKCIA 360° OBRAZU	15
1.5.1 Rovnohanná projekcia (ERP).....	15
1.5.2 Cylindrická projekcia s rovnomernou plochou (EAP).....	15
1.5.3 Kocková projekcia (CMP)	16
1.5.4 Projekcia pyramídy zrezanej gule (TSP).....	17
1.5.5 Krasterová parabolická projekcia (CPP)	17
2. ZDROJOVÉ KÓDOVANIE OBRAZU KONVENČNÝMI KODEKMI	18
2.1 JPEG.....	18
2.1.1 Kodér JPEG.....	18
2.1.2 Dekodér JPEG	19
2.2 HEIC.....	20
2.2.1 Kodér HEIC	20
2.2.2 Dekodér HEIC	21
2.3 AVIF.....	21
2.3.1 Kodér AVIF.....	22
2.3.2 Dekodér AVIF	23
2.4 JPEG XL.....	23
2.4.1 Kodér JPEG XL	23
2.4.2 Dekodér JPEG XL.....	24
2.5 VVC	24
2.5.1 Kodér VVC Intra	25
2.5.2 Dekodér VVC Intra	25
3. STROJOVÉ A HLBOKÉ UČENIE.....	26
3.1 STROJOVÉ UČENIE	26
3.1.1 Učenie s učiteľom.....	27
3.1.2 Učenie bez učiteľa.....	27
3.1.3 Učenie s odmeňovaním	27
3.2 NEURÓNOVÉ SIETE	28
3.3 HLBOKÉ UČENIE	29
3.3.1 Rekurentná neurónová sieť	30
3.3.2 Autoenkodér	30
3.3.3 Konvolučná neurónová sieť	31
3.3.4 Generatívna kompetitívna sieť	31
3.4 VYUŽITIE UČENIA V KOMPRESII OBRAZU.....	32

3.4.1	<i>Siete určené na end-to-end spracovanie</i>	32
3.4.2	<i>Siete zvyšujúce efektivitu konvenčných kodekov</i>	32
3.5	LATENTNÁ REPREZENTÁCIA OBRAZU	32
3.6	JPEG AI	32
3.7	KOMPRESIA PLANÁRNEHO A 360° OBRAZU ZALOŽENÁ NA VYUŽITÍ STROJOVÉHO A HLBOKÉHO UČENIA	34
3.7.1	<i>Full Resolution Image Compression with Recurrent Neural Networks</i>	34
3.7.2	<i>Context-adaptive Entropy Model for End-to-end Optimized Image Compression</i>	35
3.7.3	<i>High-Fidelity Generative Image Compression</i>	35
3.7.4	<i>End-to-End Optimized 360° Image Compression</i>	36
4.	OBJEKTÍVNE A SUBJEKTÍVNE METRIKY KVALITY	38
4.1	OBJEKTÍVNE METRIKY KVALITY	38
4.1.1	<i>Peak Signal to Noise Ratio</i>	38
4.1.2	<i>Structural similarity index</i>	39
4.1.3	<i>Visual information fidelity</i>	40
4.1.4	<i>Feature similarity</i>	41
4.1.5	<i>Gradient magnitude similarity deviation</i>	41
4.1.6	<i>Video multi-method assessment fusion</i>	42
4.1.7	<i>Normalized Laplacian pyramid distance</i>	43
4.2	SUBJEKTÍVNE METRIKY KVALITY	43
4.2.1	<i>Double stimulus impairment scale</i>	43
4.2.2	<i>Double stimulus continuous quality scale</i>	44
4.2.3	<i>Absolute category rating</i>	44
4.2.4	<i>Absolute category rating with hidden reference</i>	44
5.	PRAKTICKÁ ČASŤ	45
5.1	TEORETICKÝ NÁVRH FRAMEWORKU	46
5.2	DATASET OBRÁZKOV	46
5.3	PRIESTOROVÝ INDEX OBRÁZKOV	47
5.4	SPRACOVANIE, KOMPRESIA A DEKOMPRESIA KONVENČNÝMI KODEKMI	47
5.4.1	<i>Konverzia formátov obrázkov nástrojom FFmpeg</i>	47
5.4.2	<i>Kompresia kodekom JPEG</i>	49
5.4.3	<i>Kompresia kodekom HEIC</i>	50
5.4.4	<i>Kompresia kodekom AVIF</i>	51
5.4.5	<i>Kompresia kodekom JPEG XL</i>	52
5.4.6	<i>Kompresia kodekom VVC Intra</i>	52
5.5	SOFTWAREVÉ NÁSTROJE UMELEJ INTELIGENCIE	53
5.5.1	<i>Inštalácia, nastavenie a sprevádzkovanie nástroja TensorFlow</i>	54
5.5.2	<i>Inštalácia, nastavenie a sprevádzkovanie nástroja PyTorch</i>	55
5.6	KOMPRESIA OBRAZU S VYUŽITÍM ALGORITMOV STROJOVÉHO A HLBOKÉHO UČENIA	56
5.6.1	<i>Algoritmus Full Resolution Image Compression with Recurrent Neural Networks</i>	56
5.6.2	<i>Algoritmus Context-adaptive Entropy Model for End-to-end Optimized Image Compression</i>	57
5.6.3	<i>Algoritmus High-Fidelity Generative Image Compression</i>	57
5.6.4	<i>Algoritmus End-to-End Optimized 360° Image Compression</i>	58
5.7	VÝSLEDKY KOMPRESIE A DEKOMPRESIE	59
5.7.1	<i>Hardwarové nároky</i>	59
5.7.2	<i>Implementácie využívajúce CPU a GPU</i>	60

5.7.3	<i>Využitie hardwarových prostriedkov.....</i>	60
5.7.4	<i>Časy kompresie a dekompresie.....</i>	61
6.	HODNOTENIE KVALITY OBRAZU.....	65
6.1	IMPLEMENTÁCIA OBJEKTÍVNYCH METRÍK	65
6.1.1	<i>Metriky PSNR-HVS-M, MS-SSIM a VIFp.....</i>	65
6.1.2	<i>Metrika WS-PSNR.....</i>	65
6.1.3	<i>Metriky FSIMc a GMSD</i>	66
6.1.4	<i>Metrika VMAF</i>	67
6.2	APLIKÁCIA SUBJEKTÍVNEJ METRIKY ACR-HR.....	67
6.3	VÝSLEDKY OBJEKTÍVNYCH METRÍK	68
6.3.1	<i>Výsledky metriky PSNR-HVS-M.....</i>	69
6.3.2	<i>Výsledky metriky WS-PSNR</i>	70
6.3.3	<i>Výsledky metriky MS-SSIM</i>	71
6.3.4	<i>Výsledky metriky VIFp</i>	72
6.3.5	<i>Výsledky metriky FSIMc.....</i>	73
6.3.6	<i>Výsledky metriky GMSD</i>	74
6.3.7	<i>Výsledky metriky VMAF.....</i>	75
6.3.8	<i>Vyhodnotenie výsledkov porovnania jednotlivých kodekov</i>	76
6.4	VÝSLEDKY SUBJEKTÍVNEJ METRIKY ACR-HR.....	77
7.	PRAKTICKÝ NÁVRH FRAMEWORKU	78
7.1	INŠTALÁCIA.....	78
7.2	KNIŽNICE	78
7.3	OBSLUHA	79
8.	ZÁVER.....	82
	LITERATÚRA.....	84
	ZOZNAM SYMBOLOV A SKRATIEK	95
	ZOZNAM PRÍLOH.....	99

ZOZNAM OBRÁZKOV

Obrázok 1 VR headset Valve Index [2].....	13
Obrázok 2 360° kamera používaná na fotografovanie zemegule [7].....	13
Obrázok 3 2D mapa dôrazu 360° obrazu [10].....	14
Obrázok 4 ERP projekcia [12].....	15
Obrázok 5 EAP projekcia [13].....	16
Obrázok 6 CMP projekcia [14].....	16
Obrázok 7 CPP projekcia [15].....	17
Obrázok 8 Chrominančné podvzorkovanie [17].....	19
Obrázok 9 JPEG kodér a dekodér [18].....	20
Obrázok 10 Stromová štruktúra HEIC [20].....	21
Obrázok 11 Vyčítavanie frekvenčných koeficientov [22].....	22
Obrázok 12 JPEG XL kodér [23].....	24
Obrázok 13 Štruktúra umelej inteligencie [28].....	26
Obrázok 14 Štruktúra strojového učenia [30].....	27
Obrázok 15 Štruktúra modelu neurónu [33].....	28
Obrázok 16 Štruktúra hlbkej neurónovej siete [38].....	29
Obrázok 17 Štruktúra rekurentnej neurónovej siete [40].....	30
Obrázok 18 Štruktúra autoenkodéru [38].....	30
Obrázok 19 Štruktúra konvolučnej neurónovej siete [38].....	31
Obrázok 20 Štruktúra generatívnej kompetitívnej siete [44].....	31
Obrázok 21 Latentná reprezentácia obrazových dát [25].....	32
Obrázok 22 JPEG AI framework [45].....	33
Obrázok 23 Príklad rôznych hodnôt PSNR [53].....	39
Obrázok 24 Schéma metriky SSIM [58].....	40
Obrázok 25 Príklad rôznych skreslení obrazu a hodnôt SSIM indexu [59].....	40
Obrázok 26 Schéma metriky FSIMc [62].....	41
Obrázok 27 Schéma metriky GMSD [64].....	42
Obrázok 28 Schéma metriky VMAF [65].....	42
Obrázok 29 Časy kompresie.....	61
Obrázok 30 Časy kompresie pre 8K rozlíšenie.....	63
Obrázok 31 Výsledky metriky PSNR-HVS-M.....	69
Obrázok 32 Výsledky metriky WS-PSNR.....	70
Obrázok 33 Výsledky metriky MS-SSIM.....	71
Obrázok 34 Výsledky metriky VIFp.....	72
Obrázok 35 Výsledky metriky FSIMc.....	73
Obrázok 36 Výsledky metriky GMSD.....	74
Obrázok 37 Výsledky metriky VMAF.....	75
Obrázok 38 Výsledky subjektívneho hodnotenia kvality.....	77
Obrázok 39 Spustený framework.....	80
Obrázok 40 Framework po spracovaní obrázku a výpočte výsledkov.....	81

ZOZNAM TABULIEK

Tabuľka 1 Špecifikácie kodeku JPEG	18
Tabuľka 2 Špecifikácie kodeku HEIC	20
Tabuľka 3 Špecifikácie kodeku AVIF	22
Tabuľka 4 Špecifikácie kodeku JPEG XL	23
Tabuľka 5 Špecifikácie kodeku VVC Intra.....	25
Tabuľka 6 Cieľové kvality pre kompresiu pomocou metód JPEG AI	33
Tabuľka 7 Hodnotenie kvality metódou DSIS.....	44
Tabuľka 8 Hodnotenie kvality metrikou ACR.....	44
Tabuľka 9 Hardwarové a softwarové špecifikácie použitého notebooku	45
Tabuľka 10 Parametre programu SITI.....	47
Tabuľka 11 Vyžadované formáty obrázkov jednotlivých kodekov, nástrojov a metrik	48
Tabuľka 12 Parametre konverzie formátov	48
Tabuľka 13 Parametre kodéru JPEG.....	49
Tabuľka 14 Parametre dekodéru JPEG.....	49
Tabuľka 15 Parametre kodéru HEIC	50
Tabuľka 16 Parametre dekodéru HEIC.....	50
Tabuľka 17 Parametre kodéru AVIF	51
Tabuľka 18 Parametre dekodéru AVIF.....	51
Tabuľka 19 Parametre kodéru JPEG XL	52
Tabuľka 20 Parametre dekodéru JPEG XL.....	52
Tabuľka 21 Parametre kodéru VVC Intra.....	53
Tabuľka 22 Parametre dekodéru VVC Intra.....	53
Tabuľka 23 Nástroje pre TensorFlow	54
Tabuľka 24 Nástroje pre PyTorch.....	55
Tabuľka 25 Nástroje algoritmu FRICwRNN.....	56
Tabuľka 26 Parametre algoritmu FRICwRNN	56
Tabuľka 27 Nástroje algoritmu CAEMfE2EOIC	57
Tabuľka 28 Parametre algoritmu CAEMfE2EOIC.....	57
Tabuľka 29 Nástroje algoritmu HiFiC	58
Tabuľka 30 Parametre algoritmu HiFiC	58
Tabuľka 31 Nástroje algoritmu LIC360	59
Tabuľka 32 Parametre algoritmu LIC360.....	59
Tabuľka 33 Využitie hardwarových prostriedkov pri kompresii	60
Tabuľka 34 Časy kompresie	62
Tabuľka 35 Časy kompresie pre 8K rozlíšenie.....	63
Tabuľka 36 Parametre programu VQMT	65
Tabuľka 37 Parametre programu 360tools_metric	66
Tabuľka 38 Parametre programu VMAF.....	67
Tabuľka 39 Porovnanie kompresnej účinnosti metódou BD-Rate, PSNR-HVS-M.....	69
Tabuľka 40 Porovnanie kompresnej účinnosti metódou BD-Rate, WS-PSNR	70
Tabuľka 41 Porovnanie kompresnej účinnosti metódou BD-Rate, MS-SSIM	71
Tabuľka 42 Porovnanie kompresnej účinnosti metódou BD-Rate, VIFp	72
Tabuľka 43 Porovnanie kompresnej účinnosti metódou BD-Rate, FSIMc.....	73
Tabuľka 44 Porovnanie kompresnej účinnosti metódou BD-Rate, GMSD	74
Tabuľka 45 Porovnanie kompresnej účinnosti metódou BD-Rate, VMAF	75
Tabuľka 46 Knižnice potrebné pre funkčnosť frameworku.....	79

ÚVOD

V posledných rokoch sa na trhu objavili zariadenia schopné prehrávať 360° obrázky a video a získali si u spotrebiteľov veľký úspech. Jedná sa predovšetkým o náhlavné súpravy (headsety) pre virtuálnu realitu (VR) a rozšírenú realitu (AR). VR headsety sú vďaka ich konštrukcii ideálne pre sledovanie obrazu v panoramatickej 180° a 360° projekcii, v 2D aj 3D formáte.

S príchodom panoramatického obrazu bolo potrebné uchovať tento obraz v dostatočnej kvalite a pre jeho uchovanie je kvôli potrebnému vysokému rozlíšeniu tohto obrazu dôležité efektívne ho zakódovať. Navyše, s postupom času nároky na rozlíšenie a kvalitu obrazu stále rastú.

V posledných rokoch sa taktiež začala na rôzne účely masovo využívať umelá inteligencia (AI) v podobe strojového učenia (machine learning) a hlbokého učenia (deep learning). Tieto metódy je možné využiť aj pri kompresii obrazu a hodnotení kvality komprimovaného obrazu.

V práci sú okrem nových vedomostí využité aj poznatky nadobudnuté v rámci bakalárskej práce [1]. V prvej kapitole sú podané informácie o reprezentácii a charakteristikách 360° obrazu, jeho projekciách, deformáciách, tvorbe a využití. Druhá kapitola je venovaná zdrojovému kódovaniu a kompresným algoritmom prostredníctvom konvenčných kodekov. Je vysvetlený spôsob kompresie kodekmi určenými priamo pre statické snímky, akými sú JPEG a JPEG XL a video kodekmi, konkrétne možnosťami ich intra-kompresie využitej na kompresiu statických snímok, akými sú HEIC, AVIF a VVC Intra. V tretej kapitole je úvod do umelej inteligencie, strojového učenia, hlbokého učenia a neurónových sietí (neural networks). Sú v nej popísané kompresné algoritmy obrazu vychádzajúce z aktivity JPEG AI a dostupné softwarové implementácie vhodné na použitie. Vo štvrtej kapitole je popísané hodnotenie kvality obrazu pomocou vhodných objektívnych a subjektívnych metrík. Hodnotenie je dôležité na určenie kvality, s akou bol daný obrázok skomprimovaný a podáva informáciu o tom, o koľko bola znížená jeho kvalita oproti pôvodnému obrázku a či je táto kvalita dostatočná.

V piatej kapitole sa práca zaoberá praktickou časťou, ktorá sa skladá z tvorby datasetu vhodných 360° obrázkov, spracovania a kompresie obrázkov a teoretickým návrhom frameworku. Šiesta kapitola je o výsledkoch hodnotenia kvality komprimovaných obrázkov objektívnymi metódami a subjektívnym testom. V siedmej kapitole je popísaný naprogramovaný framework umožňujúci kompresiu 360° obrázkov konvenčnými metódami a metódami založenými na využití hlbokého učenia. V závere sú zhrnuté dosiahnuté výsledky a budúca práca.

1. REPREZENTÁCIA 360° OBRAZU

Obraz môže byť vytvorený viacerými spôsobmi. Prirodzený obraz je zachytený na záznamové zariadenie, ktorým môže byť fotoaparát alebo kamera. Umelý obraz je vytvorený vo vhodnom počítačovom SW, v rôznych grafických editoroch alebo programoch zameraných na 2D a 3D modelovanie. Môže to byť klasický bežný plochý obraz, alebo 360° obraz vhodný pre zobrazenie vo VR zariadeniach, ako na obrázku 1.



Obrázok 1 VR headset Valve Index [2]

1.1 Využitie 360° obrazu

V internetových databázach je možné naraziť na veľmi detailné panoramatické obrázky s rozlíšením rádovo až v gigapixeloch [3], [4], [5]. Spoločnosť Google taktiež používa na záznam zemegule kamery s 360° uhlom záberu, ako napríklad kamera na obrázku 2, z ktorých fotografie je možné pozorovať v aplikácii Google Mapy v module Street View [6]. Ďalšími možnosťami využitia je použitie v medicíne, vzdelávaní, simuláciách, tvorbe modelov a zábava, napríklad v podobe 360° videí vhodných pre VR.



Obrázok 2 360° kamera používaná na fotografovanie zemegule [7]

1.2 Charakteristiky 360° obrazu

Zorný uhol bežného plochého obrazu je obmedzený na výšku aj na šírku. 360° obraz (všesmerový, panoramatický, sférický) má zorný uhol $360^\circ \times 180^\circ$, vďaka čomu je možné zaznamenať obraz vo všetkých smeroch a uhloch naraz. Kvôli zvýšenému množstvu informácií v 360° obraze a súčasnému pozorovaniu len malého výrezu z tohto obrazu je dôležité celý tento obraz zaznamenať v dostatočne vysokom rozlíšení [8], [9].

1.3 Oblasť záujmu v 360° obraze

Pri sledovaní 360° obrazu vidí človek len malý výrez z tohto obrazu. Je dôležité určiť, na aký výrez z celého obrazu je človek zameraný najviac. Pre identifikáciu sledovaného výrezu vo VR zaradení, akým je najčastejšie VR headset (náhlavná súprava), je používané sledovanie pohybu hlavy. V pozorovanom výreze obrazu je dôležité určiť oblasť záujmu, na ktorú sa človek sústreďí najviac. Tá je určená sledovaním pohybu očí z vnútornej strany VR headsetu. Spojením týchto dvoch parametrov vznikne 2D mapa dôrazu (saliency map), akú je možné vidieť na obrázku 3. Mapy dôrazu väčšiny respondentov sú vo vysokej korelácii. Záujmová oblasť človeka sa nachádza v strede sledovaného obrazu. Ďalšou vlastnosťou ľudského zraku je sústredenie sa výrazné objekty vo videnom obraze [9].



Obrázok 3 2D mapa dôrazu 360° obrazu [10]

1.4 Deformácie 360° obrazu

Projekciou sférického obrazu do 2D roviny vznikajú v obraze deformácie. Sú to redundantné vzorky, skreslenie tvaru a nespojité hranice. Pri redundantných vzorkách sú do obrazu kódované aj neplatné pixely [11]. Nespojité hranice sú problémom, pretože na hranách obrazu, kde je obraz spájaný, vznikajú viditeľné blokové artefakty v dôsledku použitia adaptívnej kvantizácie kodéra. V prípade videosekvencií je pri skreslení tvaru znížená presnosť odhadu a kompenzácie pohybu [11]. Rôzne projekcie vnášajú do obrazu rôzne deformácie v rôzne veľkej miere. Ďalšou deformáciou je skreslenie na hraniciach

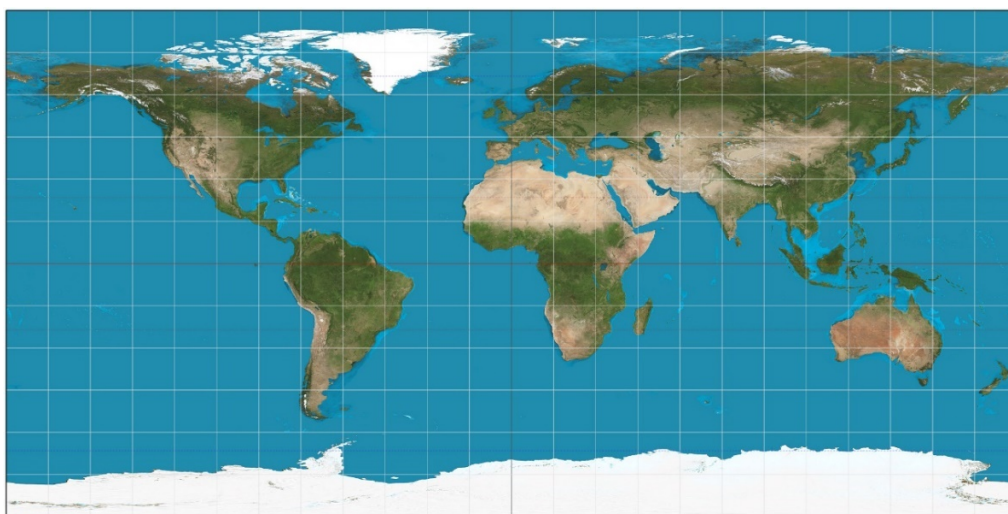
uhlov záberu vytvorené vplyvom nepresného spájania obrazov jednotlivých kamier, ktoré sú súčasťou záznamového zariadenia. Toto skreslenie je možné potlačiť alebo eliminovať presným algoritmom spájania dielčích obrazov do finálneho obrazu, alebo spracovaním vo vhodnom grafickom editore.

1.5 Projekcia 360° obrazu

Pre efektívne spracovanie 360° obrazu je potrebné tento obraz transformovať z pôvodnej všesmerovej projekcie do 2D rovinnej projekcie vhodnej na spracovanie. Pri týchto projekciách sa na hranách spracovávaného obrazu nachádzajú nespojitosti. Transformácia sférického obrazu do 2D roviny je nelineárna, čo spôsobuje jav, že pixely medzi týmito doménami spolu nekorešpondujú [11].

1.5.1 Rovnohanná projekcia (ERP)

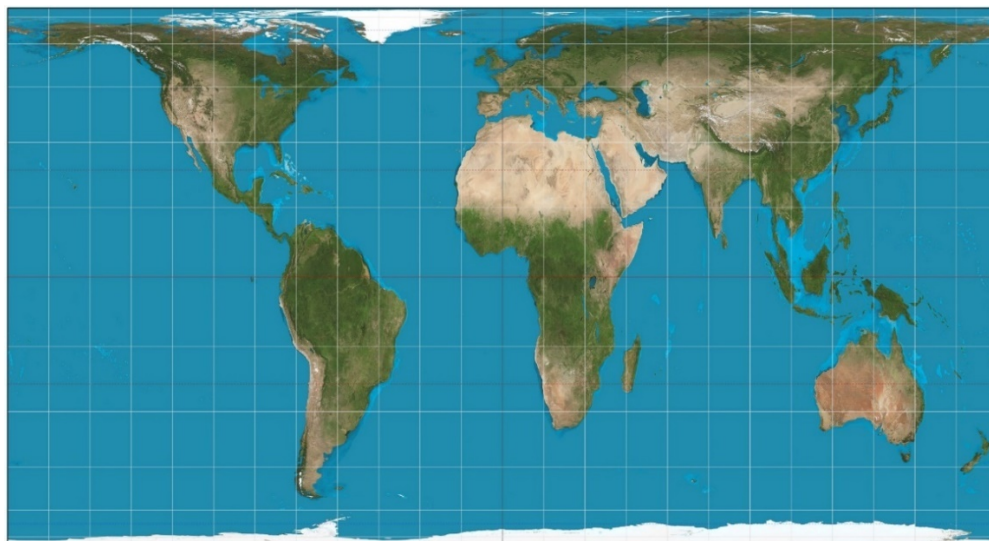
ERP je príkladom cylindrickej projekcie. Je to jedna z najjednoduchších projekcií na vytvorenie a rovnako jedna z najpoužívanejších. Jej pomer strán je 2:1. Nevýhodou ERP projekcie je deformácia v podobe veľkého množstva redundantných vzoriek a horizontálne natiahnutie v hraničnej vrchnej a spodnej oblasti obrazu [11]. Na obrázku 4 je možné vidieť príklad obrazu v ERP projekcii.



Obrázok 4 ERP projekcia [12]

1.5.2 Cylindrická projekcia s rovnomernou plochou (EAP)

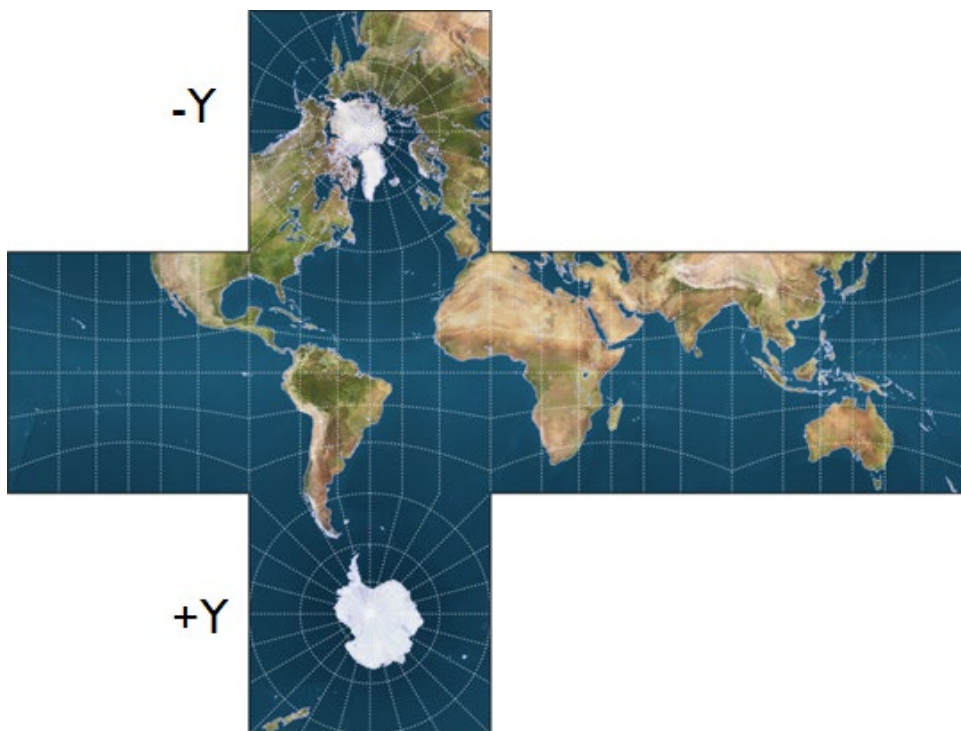
EAP je cylindrická projekcia podobná ERP, avšak je v nej odstránený problém s redundanciou vzoriek. Jej nevýhodou je ešte väčšie skreslenie tvaru, než v prípade ERP [11]. Na obrázku 5 je možné vidieť príklad obrazu v EAP projekcii.



Obrázok 5 EAP projekcia [13]

1.5.3 Kocková projekcia (CMP)

CMP je príkladom mnohostennej projekcie. Má menej geometrických deformácií ako ERP a preto je vhodnejšia na 3D vykresľovanie, čo je často používané na vykresľovanie priestoru v prostredí VR. Pri CMP projekcii je potlačené aj skreslenie tvaru. Nevýhodou je nerovnomerná distribúcia rozlíšenia, preto je pre zachovanie kvality potrebné obraz nadzorkovať [11]. Na obrázku 6 je možné vidieť príklad obrazu v CMP projekcii.



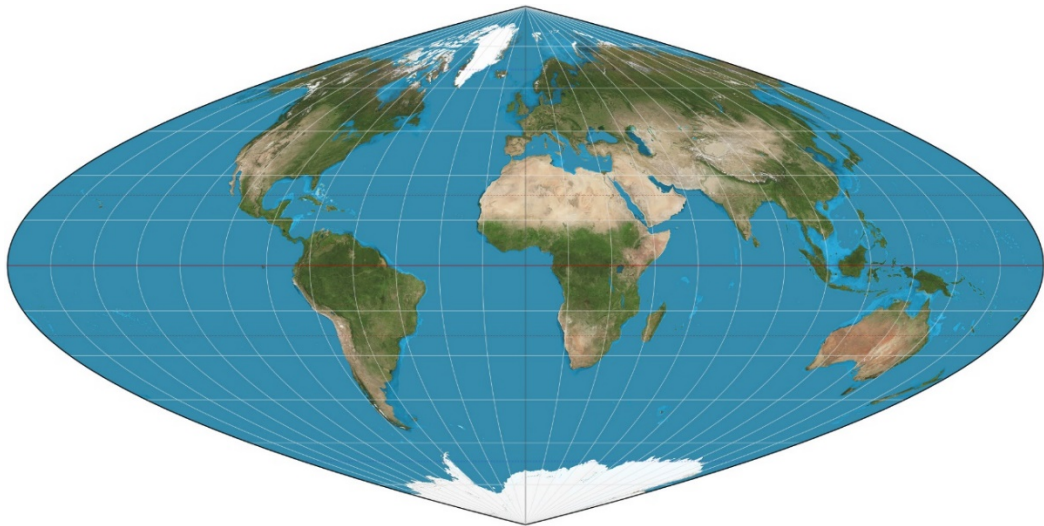
Obrázok 6 CMP projekcia [14]

1.5.4 Projekcia pyramídy zrezanej gule (TSP)

TSP je príkladom mnohostennej projekcie s nerovnomernými tvármi. Originálny sférický povrch je projektovaný do tvaru pyramídy a preskupený do obdĺžnika. Spodná časť pyramídy je vzorkovaná v plnom rozlíšení a strany pyramídy sú vzorkované v zníženom rozlíšení, čím nevznikajú redundantné vzorky. Táto projekcia je vhodná pre streamovanie [11].

1.5.5 Krasterová parabolická projekcia (CPP)

CPP je príkladom pseudocylindrickej projekcie. Je využívaná pri posudzovaní kvality pre jej charakteristiku rovnomerne rozdelenej plochy. Výhodou tejto projekcie je transformácia sféry do 2D roviny bez zmeny priestorového rozlíšenia [11]. Na obrázku 7 je možné vidieť príklad obrazu v CPP projekcii.



Obrázok 7 CPP projekcia [15]

2. ZDROJOVÉ KÓDOVANIE OBRAZU KONVENČNÝMI KODEKMI

V nekomprimovanom obraze je obsiahnuté veľké množstvo dát, ktoré sú náročné na prenos. Pri použití 360° reprezentácie obrazu je toto množstvo dát ešte väčšie vplyvom veľmi vysokého rozlíšenia potrebného pre zachovanie kvality. Z tohoto dôvodu je nutné znížiť objem dát bez pozorovateľného zníženia kvality obrazu. Zníženie objemu prenášaných dát je možné dosiahnuť zdrojovým kódovaním. Pri kódovaní je odstránená redundancia obrazového signálu (bezstratová kompresia) a irelevantné dáta (stratová kompresia). Pomocou stratovej kompresie je možné dosiahnuť vyšší kompresný pomer, ale nevýhodou je zníženie kvality obrazu. Dnes používané obrazové kodeky umožňujú použiť samostatne bezstratovú kompresiu, alebo kombináciu bezstratovej a stratovej kompresie. Digitálny obrazový signál je zložený z pixelov, kde každý pixel má 3 farebné zložky – červenú, zelenú a modrú (sústava RGB) [1], [8].

2.1 JPEG

Kodek bol vyvinutý v roku 1992 organizáciami ISO a CCITT a je najviac používaným stratovým kodekom pre kódovanie fotografií, obrázkov a grafiky, avšak je ním možné kódovať obraz aj bezstratovo. Kompresný pomer môže dosiahnuť úroveň až 1:10 pri zachovaní dostatočnej kvality obrazu. Bitová hĺbka je 8 bitov pre každý kanál [16]. V tabuľke 1 sú zobrazené základné špecifikácie kodeku JPEG.

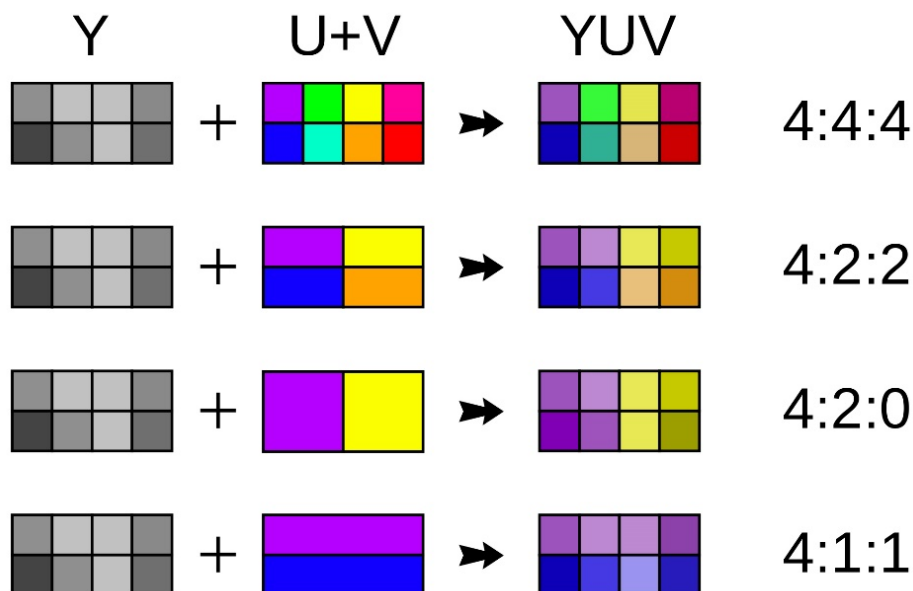
Tabuľka 1 Špecifikácie kodeku JPEG

Maximálne rozlíšenie	65535×65535
Kompresia	stratová, bezstratová
Bitová hĺbka	v základe 8 bitov, neskôr 10 až 12 bitov
Farebné podvzorkovanie	4:0:0, 4:2:0, 4:2:2, 4:4:4
Veľkosť blokov	8×8
Pomer strán blokov	1:1
Entropické kódovanie	RLE, Huffmanovo
Podpora HDR	Nie
Podpora WCG	Nie
Podpora alfa kanálu	Nie

2.1.1 Kodér JPEG

Pri kompresii je obraz prevedený zo sústavy RGB do sústavy YUV. YUV signál obsahuje luminančnú zložku Y a rozdielové chrominančné zložky U a V. Chrominančný signál je podvzorkovaný z režimu 4:4:4, kde pre každú vzorku Y existuje 1 vzorka U a 1 vzorka V, do režimu 4:2:0, kde existuje 1 U a 1 V vzorka pre každú štvoricu vzoriek Y. Tým je

rozlíšenie chrominančného signálu znížené na štvrtinu a celkový objem dát je zredukovaný na polovičnú veľkosť. Na obrázku 8 je možné vidieť rôzne režimy chrominančného podvzorkovania.

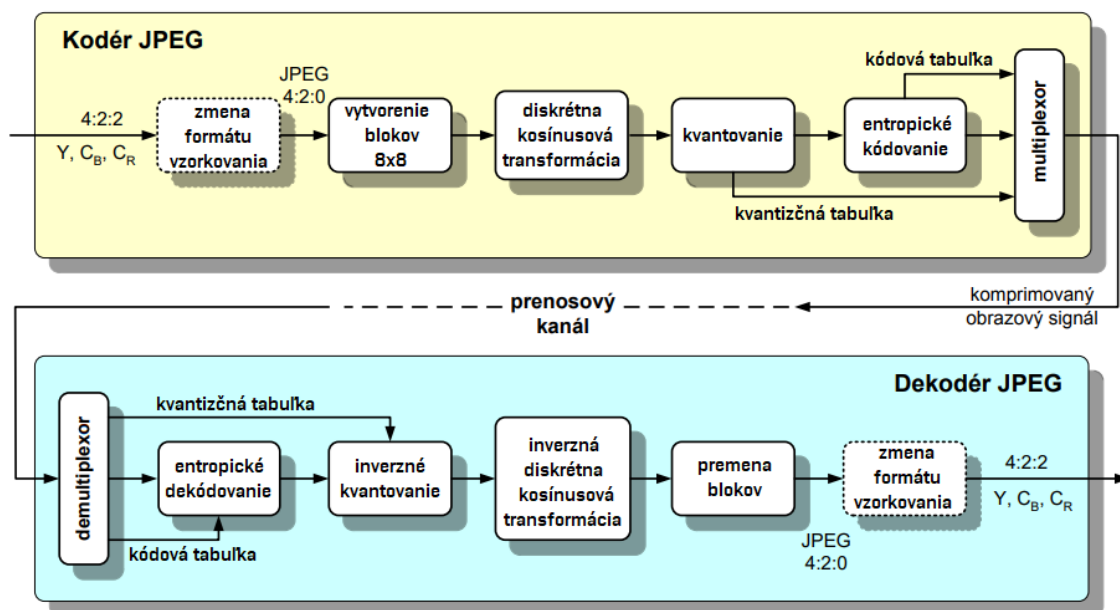


Obrázok 8 Chrominančné podvzorkovanie [17]

Jednotlivé kanály Y, U a V sú rozdelené na bloky o veľkosti 8×8 pixelov. Obráz je rozdelený na MCU. V jednom MCU sa nachádzajú 4 bloky Y, 1 blok U a 1 blok V. Jednotlivé bloky sú prevedené do frekvenčnej domény pomocou 2D-DCT. Frekvenčné koeficienty sú vydelené kvantizačnou tabuľkou. Vysokofrekvenčné koeficienty s hodnotou blízko 0 ďalej kódované nie sú. Frekvenčné koeficienty sú vyčítavané z bloku diagonálne a kódované RLE algoritmom a Huffmanovým kódom [16].

2.1.2 Dekodér JPEG

Dáta sú entropicky dekódované, bloky frekvenčných koeficientov sú vynásobené kvantizačnou tabuľkou a pomocou IDCT sú prevedené späť do priestorovej domény. YUV signál je prevedený na signál RGB a týmto je celý obraz dekódovaný. Na obrázku 9 je zobrazená bloková schéma kodéru a dekodéru JPEG.



Obrázok 9 JPEG kodér a dekodér [18]

2.2 HEIC

Kodek bol vyvinutý v roku 2015 organizáciou MPEG a odvodený od video kodeku HEVC. Využíva intra kompresiu kodeku HEVC a je o polovicu úspornejší ako kodek JPEG. Okrem statických snímok podporuje aj kódovanie sekvencie snímok s využitím inter predikcie. Podporuje stratovú aj bezstratovú kompresiu, 8/10-bitovú kvantizačnú hĺbku na kanál, alfa kanál, hĺbkovú mapu, SDR a HDR dynamické rozsahy a obrazové derivácie pre nedeštruktívnu editáciu [19]. V tabuľke 2 sú zobrazené základné špecifikácie kodeku HEIC.

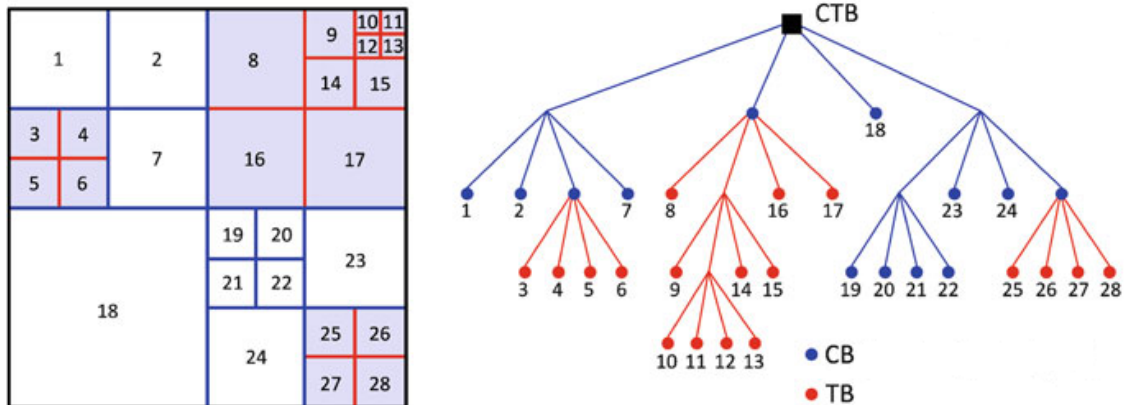
Tabuľka 2 Špecifikácie kodeku HEIC

Kompresia	stratová, bezstratová
Bitová hĺbka	8, 10, 12, 16 bitov
Farebné podvzorkovanie	4:0:0, 4:2:0, 4:2:2, 4:4:4
Veľkosť blokov	4×4 až 64×64
Pomer strán blokov	1:1, 1:2, 2:1
Entropické kódovanie	RLE, CABAC
Podpora HDR	Áno
Podpora WCG	Áno
Podpora alfa kanálu	Áno

2.2.1 Kodér HEIC

Obraz je rozdelený na kódové stromové jednotky (CTU), ktoré sú zložené z kódových stromových blokov (CTB) pre jasový aj chrominančný signál. CTB sa delia na menšie

kódové jednotky (CB). Spojením jasového a chrominančných CB vznikne kódová jednotka (CU). CU môžu byť predikčné jednotky (PU) alebo transformačné jednotky (TU). Na obrázku 10 je zobrazená stromová štruktúra kodeku HEIC.



Obrázok 10 Stromová štruktúra HEIC [20]

V prípade statického obrázku sú všetky bloky typu TU a sú kódované pomocou intrapredikcie, kde je dostupných 35 predikčných módov. Následne sú pomocou DCT transformované do frekvenčnej domény, v prípade TU s veľkosťou 4×4 pomocou DST. V prípade syntetickej grafiky je často výhodnejšie transformáciu nepoužiť. Ďalším krokom je kvantizácia, ktorá je nelineárna. Koefficienty sú vyčítané diagonálne a entropicky zakódované pomocou RLE a CABAC algoritmov [19].

2.2.2 Dekodér HEIC

Dáta sú entropicky dekodované, bloky frekvenčných koeficientov sú vynásobené kvantizačnou tabuľkou a pomocou IDCT, prípadne IDST sú prevedené späť do priestorovej domény. Deblokovacím a SAO filtrom sú vyhladené hrany a skreslenia viditeľné na hraniciach jednotlivých transformačných blokov. YUV signál je prevedený na signál RGB.

2.3 AVIF

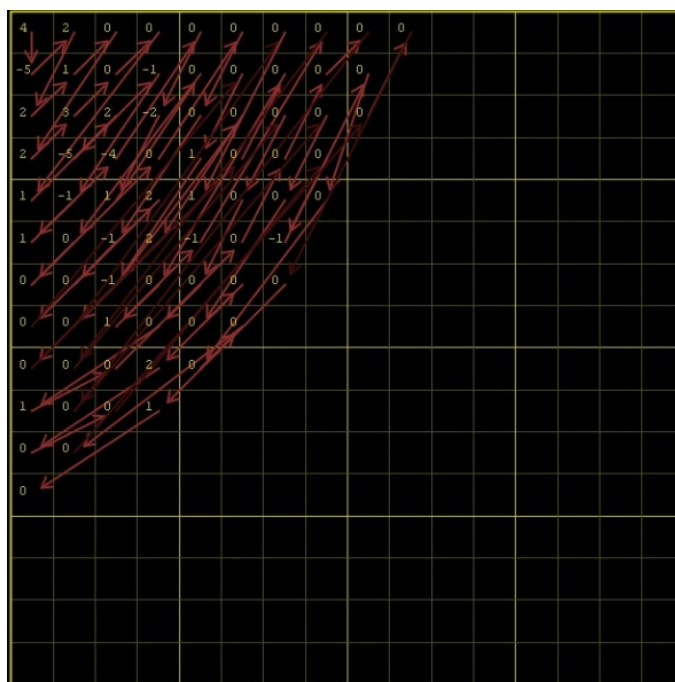
Kodek bol vyvinutý v roku 2019 organizáciou AOMedia a odvodený od video kodeku AV1. Je konkurenciou kodeku HEIC. Využíva kompresné algoritmy kodeku AV1. Podporuje stratovú aj bezstratovú kompresiu, 8/10/12-bitovú kvantizačnú hĺbku na kanál, alfa kanál, hĺbkovú mapu, SDR a HDR dynamické rozsahy, WCG, filmový šum, animácie a obrazové sekvencie [21]. V tabuľke 3 sú zobrazené základné špecifikácie kodeku AVIF.

Tabuľka 3 Špecifikácie kodeku AVIF

Kompresia	stratová, bezstratová
Bitová hĺbka	8, 10, 12 bitov
Farebné podvzorkovanie	4:0:0, 4:2:0, 4:2:2, 4:4:4
Veľkosť blokov	4×4 až 128×128
Pomer strán blokov	1:1, 1:2, 2:1, 1:4, 4:1
Entropické kódovanie	RLE, MSBAC
Podpora HDR	Áno
Podpora WCG	Áno
Podpora alfa kanálu	Áno

2.3.1 Kodér AVIF

Obraz je rozdelený na superbloky 128×128 pixelov, ktoré sú ďalej podľa potreby zmenšené na bloky s nižším rozlíšením a rôznym pomerom strán. Pomocou deltakódovania je pri intrapredikcii možné použiť 56 smerových predikčných módov. Ďalej je v rámci predikcie využitá aj predikcia farby z jasú (chroma from luma prediction), pretože blízke vzorky obrazu sú vo vzájomnej korelácii. Pri transformačnom kódovaní sú použité algoritmy DCT, ADST, FlipADST (zrkadlovo prevrátená ADST) a v prípade syntetického obrazu transformácia použitá nie je. V horizontálnom aj vertikálnom smere je možné použiť naraz rôzne transformácie a vzájomne ich kombinovať. Kvantizácia môže byť lineárna alebo nelineárna. Koeficienty sú vyčítané diagonálne v tvare štvrtkruhu a entropicky zakódované RLE a MSBAC algoritmami [21]. Na obrázku 11 je zobrazené vyčítavanie frekvenčných koeficientov kodeku AVIF.



Obrázok 11 Vyčítavanie frekvenčných koeficientov [22]

2.3.2 Dekodér AVIF

Dáta sú entropicky dekodované, bloky frekvenčných koeficientov sú vynásobené kvantizačnou tabuľkou a pomocou inverzných transformácií prevedené späť do priestorovej domény. Po deblokovanom filtri je použitý CDEF filter, ktorým sú vyhladené hrany v určitých smeroch ďalej zjemnené. Po prevedení z YUV do RGB je na obraz aplikovaný syntetizátor filmového šumu, ktorého úlohou je vnieŕ do obrazu vysokofrekvenčné zložky podobné tým, ktoré boli odstránené pri kompresii.

2.4 JPEG XL

Kodek bol vyvinutý v roku 2022 organizáciou JPEG za účelom efektívnejšej redukcie dát pri kompresii statického obrazu a predpokladá sa, že nahradí kodek JPEG. Podporuje stratovú aj bezstratovú kompresiu. Kompresný pomer je v priemere o polovicu lepší ako pri kodeku JPEG pri zachovaní rovnakej kvality obrazu. Oproti JPEG obsahuje radu nových funkcií, ako 32-bitovú kvantizačnú hĺbku na kanál, vyššie maximálne rozlíšenie, možnosť kódovať sekvenciu obrázkov, podporu alfa kanálu, vrstvy, obdĺžnikové transformačné makrobloky, predikciu farby z jasu, DCT s variabilnou veľkosťou makroblokov, adaptívnu kvantizáciu, podporu farebného priestoru XYB, generátor šumu, filtre pre potlačenie blokových artefaktov, kódovanie grafiky a animácii, široký farebný priestor (WCG) a vysoký dynamický rozsah (HDR). Umožňuje deliť obraz na prúžky a výrezy obrazu kódovať samostatne, zakódovať samostatne znaky textu a používať ich pri predikcii. Pre zrýchlenie spracovania umožňuje multivláknové kódovanie a dekodovanie [23]. V tabuľke 4 sú zobrazené základné špecifikácie kodeku JPEG XL.

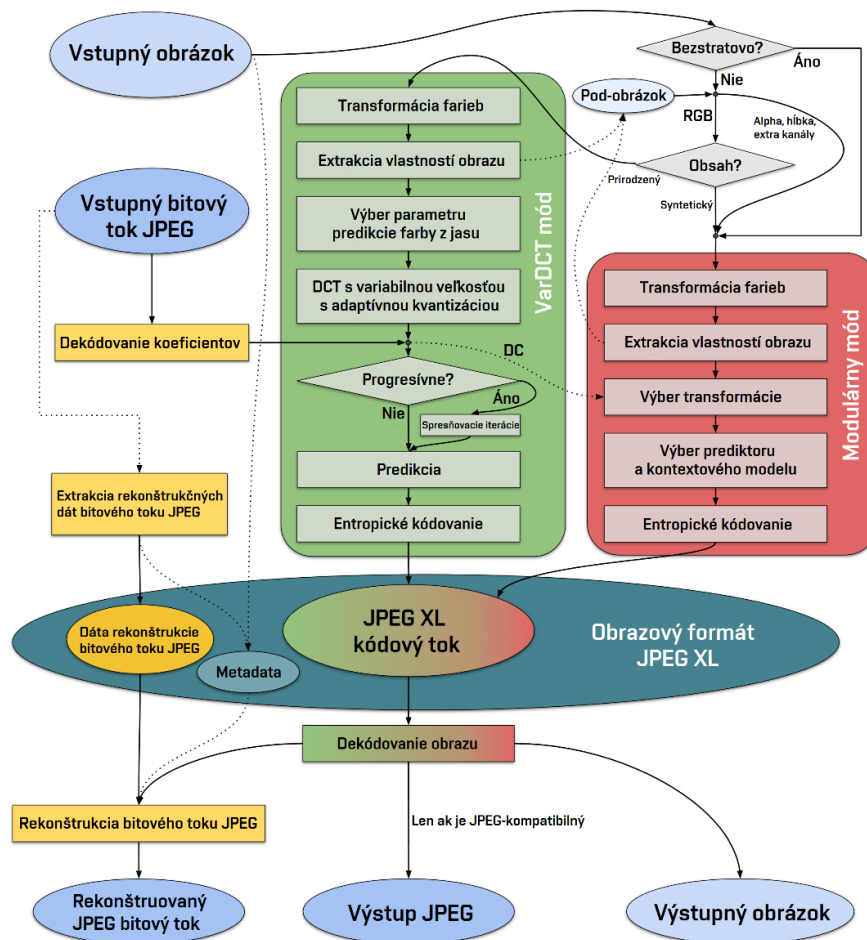
Tabuľka 4 Špecifikácie kodeku JPEG XL

Kompresia	stratová, bezstratová
Bitová hĺbka	8, 10, 12, 16, 32 bitov
Farebné podvzorkovanie	4:0:0, 4:2:0, 4:2:2, 4:4:4
Veľkosť blokov	2×2 až 256×256
Entropické kódovanie	Huffmanovo, ANS, LZ77
Podpora HDR	Áno
Podpora WCG	Áno
Podpora alfa kanálu	Áno
Podpora 360° obrazu	Áno

2.4.1 Kodér JPEG XL

Hlavnou novinkou je delenie obrazu na transformačné bloky s variabilnou veľkosťou a pomerom strán 1:1, 1:2 a 1:4 v horizontálnom aj vertikálnom smere pomocou VarDCT, Veľkosť blokov je 8×8 až 256×256 pixelov. V prípade potreby sú bloky 8×8 zmenšené až na veľkosť 2×2. V rámci intrapredikcie je použitá predikcia farebných vzoriek

z jasových vzoriek. Pre potlačenie kompresných artefaktov sú hrany blokov v kodéri zaostrené pomocou Gabor transformácie. Na bloky je aplikovaná DCT a koeficienty sú kódované adaptívnou kvantizáciou. Diagonálne vyčítané koeficienty sú entropicky zakódované Huffmanovým alebo ANS kódovaním a dodatočne je možné použiť bezstratový kompresný algoritmus LZ77 [23]. Na obrázku 12 je zobrazená bloková schéma kodeku JPEG XL.



Obrázok 12 JPEG XL kodér [23]

2.4.2 Dekodér JPEG XL

Dáta sú entropicky dekódované, bloky frekvenčných koeficientov sú vynásobené kvantizačnou tabuľkou a pomocou IDCT sú prevedené späť do priestorovej domény. Pre vyhladenie hrán a potlačenie blokových artefaktov sú okraje blokov jemne rozostrené spätnou Gabor transformáciou. YUV alebo XYB signál je prevedený na signál RGB.

2.5 VVC

Kodek bol predstavený v roku 2020 a slúži primárne na kódovanie videa. Popritom je ním možné komprimovať obrázky s využitím jeho intra-kompresie. Je o 30 – 50%

úspornejší ako kodek HEIC/HEVC. Keďže je pomerne nový, stále prebiehajú práce na dokončení všetkých jeho funkcií a optimalizácii rýchlosti kódovania. Má podporu pre stratovú aj bezstratovú kompresiu, 8/10/12-bitovú kvantizačnú hĺbku na kanál, alfa kanál, hĺbkovú mapu, SDR a HDR dynamické rozsahy, WCG a efektívnejšie kódovanie 360° obrázkov a videa [24]. V tabuľke 5 sú zobrazené základné špecifikácie kodeku VVC Intra.

Tabuľka 5 Špecifikácie kodeku VVC Intra

Kompresia	stratová, bezstratová
Bitová hĺbka	8, 10, 12, 16, 32 bitov
Farebné podvzorkovanie	4:0:0, 4:2:0, 4:2:2, 4:4:4
Veľkosť blokov	4×4 až 128×128
Entropické kódovanie	CABAC
Podpora HDR	Áno
Podpora WCG	Áno
Podpora alfa kanálu	Áno
Podpora 360° obrazu	Áno

2.5.1 Kodér VVC Intra

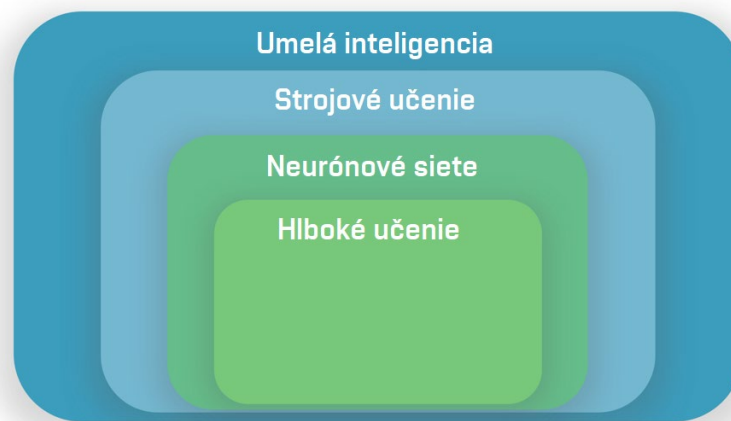
Po chromatickom podvzorkovaní je obraz rozdelený na bloky 128×128 a je vytvorená CTB štruktúra podobná kodeku HEIC a AVIF. Z CTU jednotiek sú vytvárané dlaždice, pružky a subsnímky a jednotlivé dlaždice je možné spracovávať paralelne a rôzne subsnímky môžu mať rozdielnu úroveň nastavenia kvality a ďalších parametrov. CTU je možné okrem vertikálneho a horizontálneho smeru rozdeliť aj diagonálne. Na spresnenie predikcie sú použité obojsmerné vektory toku, kde je presnosť zvýšená druhýkrát na strane dekodéra. Pre intrapredikciu je dostupných 65 predikčných smerov. Ďalej je použitá predikcia chrominančných zložiek z luminančnej zložky, ako aj u AVIF. Na transformáciu CU do frekvenčnej domény je použitá DCT a DST, respektíve kopírovanie blokov bez transformácie v prípade umelého obsahu. Adaptívnou závislou kvantizáciou je možné použiť jednu z dvoch kvantizačných tabuliek pre jeden kvantizačný parameter a tým zvýšiť presnosť frekvenčných koeficientov. Pri kódovaní 360° obrazu sa počíta so súvislým prechodom medzi hranami obrazu, čím sú potlačené hranové artefakty pri sledovaní obrazu vo VR. Pre entropické kódovanie je použitý algoritmus CABAC [24].

2.5.2 Dekodér VVC Intra

Dáta sú entropicky dekodované, bloky sú vynásobené príslušnými kvantizačnými tabuľkami a prevedené do časovej domény aplikáciou IDCT a IDST. Výhodou je možnosť dekodovať len určité dlaždice, vďaka čomu je kodek veľmi efektívny v 360° a VR aplikáciách. V dekodéri sú spresnené vektory toku a na zníženie kompresných artefaktov je použitý deblokovací filter [24].

3. STROJOVÉ A HLBOKÉ UČENIE

Umelá inteligencia (AI) bola vytvorená za účelom riešenia úloh strojom tak, ako tieto úlohy rieši človek. Základy AI boli položené ešte v 20. storočí, no najväčší rozmach zažíva práve v posledných rokoch. Je to aj vďaka rapídne nárastu dostupného výpočtového výkonu. AI je dnes súčasťou mnohých zariadení, akými sú počítače, smartfóny, autá, roboty alebo digitálni asistenti a je využívaná na rôzne účely, akými sú spracovanie obrazu a zvuku, počítačové videnie, klasifikácia dát, bezpečnosť, autonómne riadenie, výcvik armády alebo diagnostika pacientov v medicíne [25], [26], [27]. Zároveň je možné využiť AI pre úlohy spracovania a kompresie 360° obrazu, na čo sú použité aj konvenčné metódy a kodeky. Na obrázku 13 sú znázornené spojitosti medzi umelou inteligenciou, strojovým učením, neurónovými sieťami a hlbokým učením.

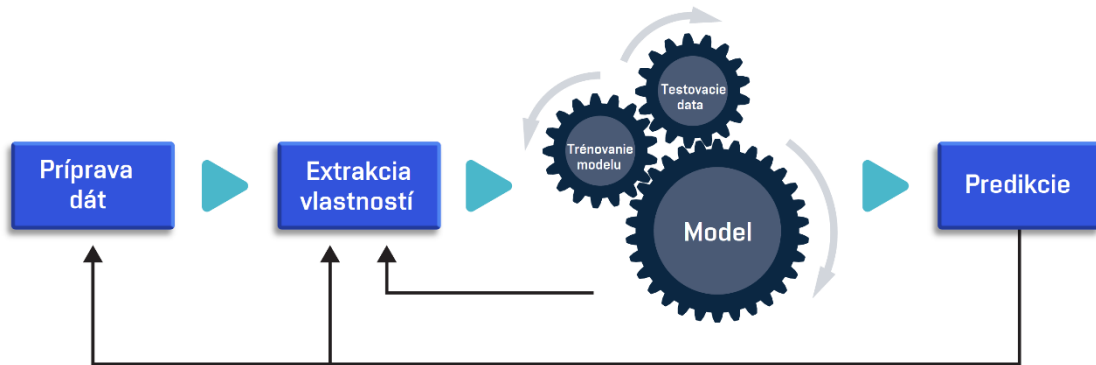


Obrázok 13 Štruktúra umelej inteligencie [28]

3.1 Strojové učenie

Strojové učenie (ML) je oblasťou AI, ktorá sa zaoberá algoritmami a technikami umožňujúcimi PC systému učiť sa. Výsledkom je, že stroj dokáže vykonať úlohu bez exaktne zadaných vstupných parametrov. Stroj sa učí tým, že postupne zlepšuje svoj výsledok opakovaným vykonávaním úlohy a využíva pri tom nadobudnuté skúsenosti. Dôležité je, aby sa stroj dokázal učiť samostatne. Učenie prebieha formou vytvorenia matematického modelu, ktorý je tréňovaný na tréningovom datasete. Štatistika je v ML využívaná na tvorbu predpovedí, ktoré sú súčasťou vstupných informácií a je dôležitá pre správne rozhodovanie. Natréňovaný model je potom možné aplikovať na riešenie úloh s rôznymi inými vstupnými dátami rovnakého typu. Príkladom môže byť automatické rozpoznávanie fotografovaných scén. Úlohy, ktoré je možné riešiť s využitím ML, sú rozdelené do troch kategórií: klasifikácia, regresia a zhukovanie. Klasifikácia dát je rozdelenie dát do tried. Regresia je odhadovanie hodnoty výstupu na základe hodnoty

vstupu. Zhukovanie je zaradovanie dát do skupín s podobnými vlastnosťami. Modelov strojového učenia je viacero, najznámejšími z nich sú: rozhodovacie stromy, algoritmus k-najbližších susedov, podporné vektory alebo lineárna regresia a neurónové siete. Existujú tri druhy algoritmov strojového učenia: učenie s učiteľom, učenie bez učiteľa a učenie formou odmeňovania [25], [26], [27], [29]. Na obrázku 14 je znázornená štruktúra strojového učenia.



Obrázok 14 Štruktúra strojového učenia [30]

3.1.1 Učenie s učiteľom

Základom je dostatočne veľký tréningový dataset, obsahujúci klasifikované dvojice vstupu a výstupu. Model je tréningovaný tak, že algoritmus sa snaží zo vstupných dát vytvoriť odpovedajúce výstupné dáta. Toto učenie je silno závislé na dostatočne veľkom množstve kvalitných dát [29].

3.1.2 Učenie bez učiteľa

Učenie bez učiteľa je používané pri neklasifikovaných dátach, pre ktoré nie sú k dispozícii odpovedajúce dvojice vstupných a výstupných dát. Algoritmy vyhľadávajú v dátach určité vzorce a na základe týchto vzorcov je možné dáta roztriediť do skupín s určitými vlastnosťami a vzťahmi medzi nimi. Učenie bez učiteľa a s učiteľom je možné kombinovať [31].

3.1.3 Učenie s odmeňovaním

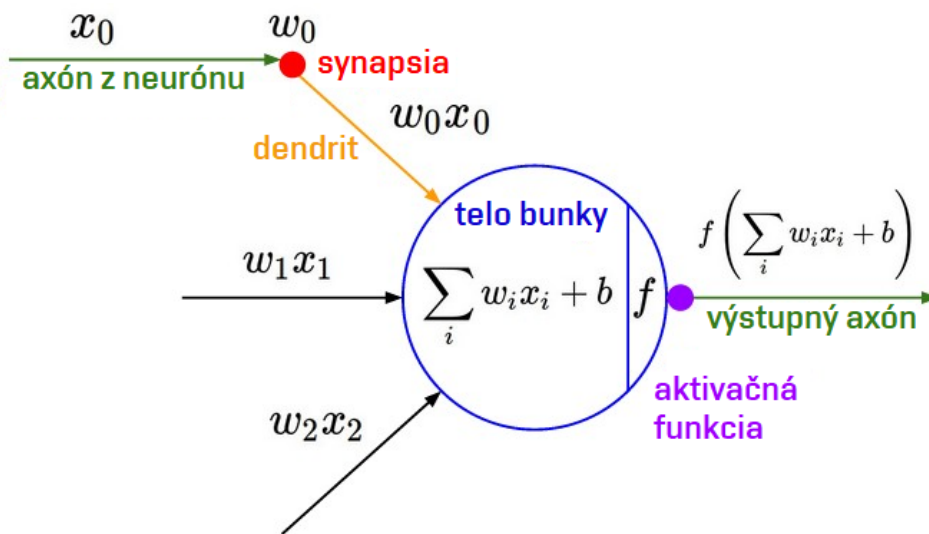
Učenie s odmeňovaním je odlišné od učenia s učiteľom a bez učiteľa. Základom je vytvorenie takzvaného agenta, ktorý sa učí prostredníctvom interakcie s prostredím. Tieto pravidlá musia byť vopred definované a označujú operácie, ktoré môže agent vykonávať. Ďalej musí byť definovaná odmeňovacia funkcia, pomocou ktorej je možné vyhodnotiť, či bolo dané rozhodnutie agenta prospešné alebo nie. Agent testuje rôzne možnosti metódou pokus-omyl a postupne sa učí, ako sa rozhodovať v konkrétnych situáciách [32].

3.2 Neurónové siete

Neurónové siete (NN), celým názvom umelé neurónové siete (ANN), sú oblasťou ML zameranou na modelovanie funkcie ľudského mozgu. Ľudský mozog je zložený z neurónov. Súčasťou neurónov sú axóny a dendrity. Axóny fungujú ako vstup do neurónu a neurón ich môže obsahovať viacero. Dendrity fungujú ako výstup neurónu a každý neurón obsahuje presne jeden dendrit. Neuróny sú medzi sebou prepojené synapsiami. Aby bol neurón aktívny, musí byť prekročená jeho aktivačná úroveň. Umelý neurón je potom možné modelovať matematicky rovnicou (1) ako:

$$Y = f\left(\sum_i x_i w_i + b\right), \quad (1)$$

kde Y je výstup neurónu, f je prenosová funkcia neurónu, x_i sú vstupy neurónu, w_i sú synaptické váhy označujúce dôležitosť príslušných vstupov a b je prahová aktivačná hodnota neurónu [25], [26]. Na obrázku 15 je zobrazená štruktúra takto modelovaného neurónu.



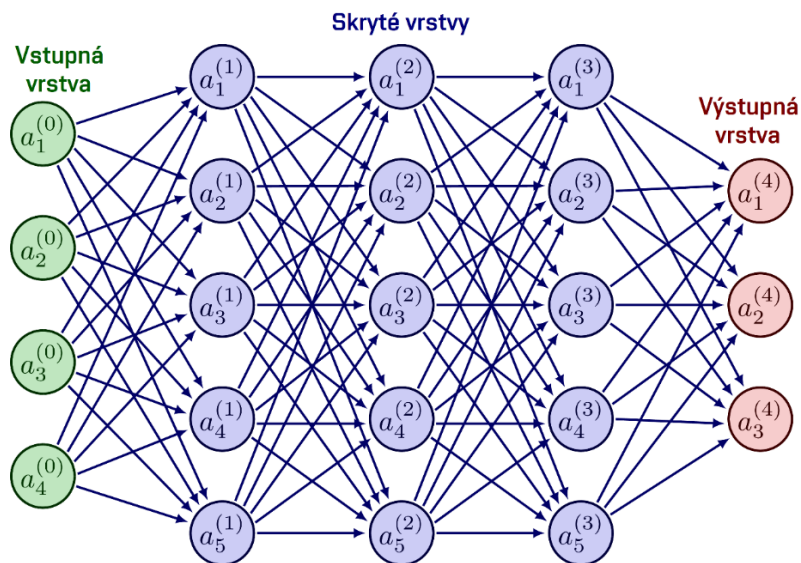
Obrázok 15 Štruktúra modelu neurónu [33]

Neurónová sieť je modelovaná prepojením neurónov. Neuróny sú rozdelené do troch druhov vrstiev. Medzi vstupnou a výstupnou vrstvou sa nachádza skrytá vrstva. Existuje niekoľko architektúr neurónových sietí, každá má svoju prenosovú funkciu. Najjednoduchšia architektúra je perceptron. Túto architektúru predstavuje samostatný neurón a prenosová funkcia tejto architektúry je skoková. Perceptron funguje ako binárny klasifikátor. Zložitejšia architektúra je viacvrstvový perceptron (MLP). Je to plne prepojená dopredná (FF) NN. FF znamená, že výstup každého neurónu jednej vrstvy je prepojený so všetkými vstupmi neurónov nasledujúcej vrstvy. Táto architektúra je zložená z troch vrstiev perceptrónov a obsahuje vstupnú vrstvu, jednu skrytú vrstvu

a výstupnú vrstvu. Prenosová funkcia MLP architektúry má tvar sigmoidy alebo hyperbolickej tangenty. MLP je možné využiť ako aproximátor ľubovoľnej funkcie. K učeniu MLP je využitá technika učenia s učiteľom, takzvané spätné šírenie chyby (error backpropagation) [25], [26].

3.3 Hlboké učenie

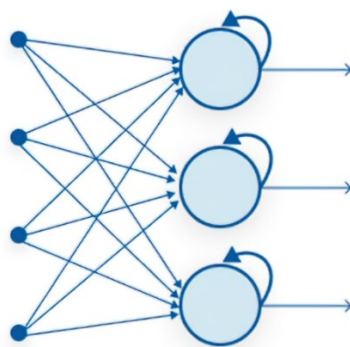
Hlboké učenie (DL) je oblasťou neurónových sietí s rozšírenými možnosťami architektúr a funkcionalít. Rozdiel od obyčajnej ANN je využitie viac ako jednej skrytej vrstvy a takto vytvorené siete sa nazývajú hlboké neurónové siete (DNN). Na trénovanie pomocou DL je potrebné omnoho väčšie množstvo dát, ako v prípade ML. Ďalším rozdielom oproti ML je nezávislosť DL na ľudskej interakcii. Trénovanie pomocou DL trvá značne dlhšie, ako v prípade ML, ale konečné výsledky sú presnejšie. Zatiaľ čo pomocou ML je možné robiť len jednoduché a lineárne operácie, pomocou DL je z podstaty možné robiť komplexnejšie a nelineárne operácie a je ich možné dobre paralelizovať pri využití špecializovaného HW [9], [25], [34], [35]. Nelinearita operácií je dosiahnutá pomocou aktivačných vrstiev, ktorých prenosové funkcie sú okrem už spomenutých aj často používané ReLU, Leaky ReLU alebo ELU [36] a mnohé ďalšie. Ďalšie druhy vrstiev sú združovacie vrstvy realizujúce podvzorkovanie a zahadzovacie vrstvy realizujúce zahadzovanie určitých vzoriek, aby nedochádzalo k pretrénovaniu siete. Vďaka tomu je daná sieť flexibilnejšia, je znížený počet parametrov siete a výpočtová náročnosť je nižšia [37]. Existuje veľké množstvo architektúr DNN, niektoré základné architektúry sú popísané ďalej. Na obrázku 16 je znázornená štruktúra základnej DNN.



Obrázok 16 Štruktúra hlbokkej neurónovej siete [38]

3.3.1 Rekurentná neurónová sieť

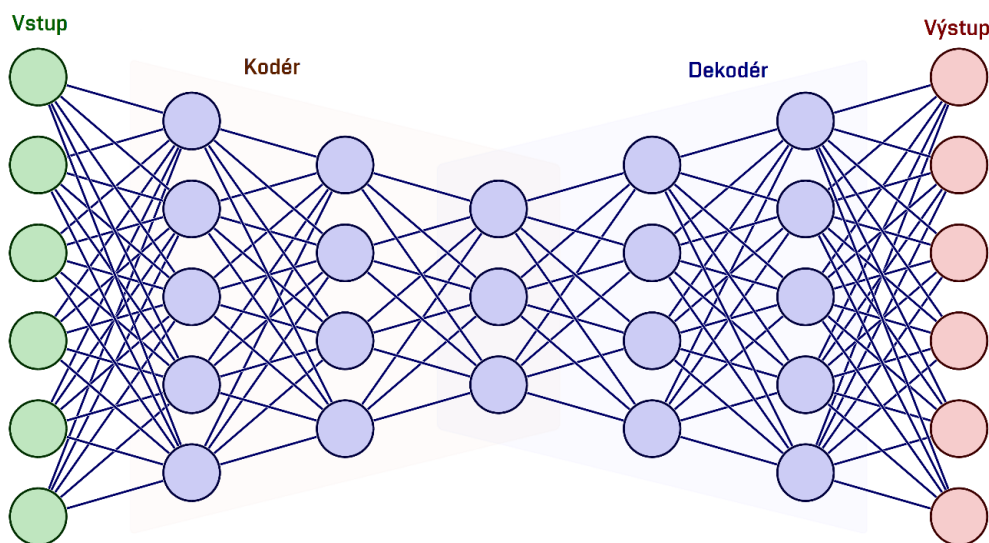
Rekurentná neurónová sieť (RNN) je DNN s pridaným prepojením neurónov v rámci jednej vrstvy, vďaka čomu je vytvorená pamäť a tým je možné navyše spracovať časové údaje. Existujú aj sofistikovanejšie RNN siete, ako LSTM alebo GRU siete. Schopnosťou LSTM siete je učenie sa dlhodobějších závislostí. GRU sieť je podobná LSTM, ale je v nej definovaných menej parametrov, vďaka čomu je rýchlejšia pri použití na menších a menej častých súboroch dát [39]. Na obrázku 17 je zobrazená štruktúra rekurentnej neurónovej siete.



Obrázok 17 Štruktúra rekurentnej neurónovej siete [40]

3.3.2 Autoenkodér

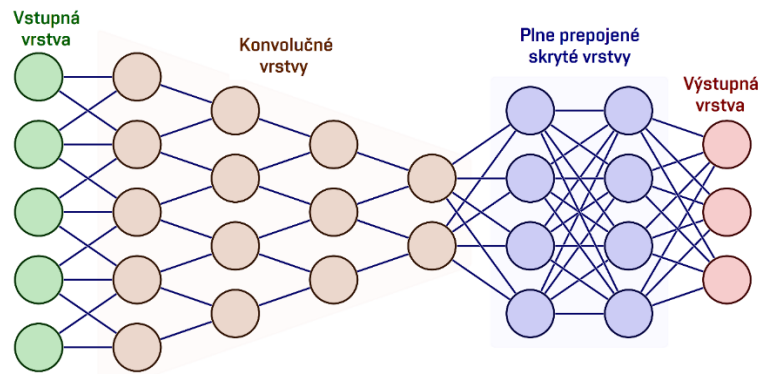
Autoenkodér (AE) je DNN obsahujúcou vrstvu s nižšou priepustnosťou, okolo ktorej sú symetricky usporiadané ďalšie skryté vrstvy. Vstupnou časťou je realizovaná kompresia obrazu a výstupnou časťou následná dekompresia obrazu [25], [41]. Na obrázku 18 je znázornená štruktúra autoenkodéru.



Obrázok 18 Štruktúra autoenkodéru [38]

3.3.3 Konvolučná neurónová sieť

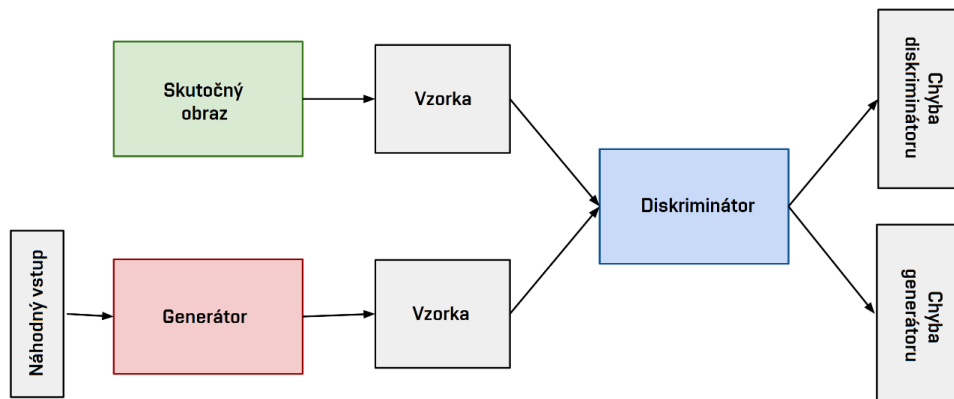
Konvolučná neurónová sieť (CNN) je DNN, v ktorej sú medzi vstupnú vrstvu a skupinu skrytých vrstiev vložené konvolučné vrstvy. Neuróny konvolučnej vrstvy obsahujú dva vstupy, čím je možné tieto vstupy konvolvovať a jeden alebo dva výstupy slúžiacie ako vstupy neurónov nasledujúcej vrstvy. Skryté vrstvy sú využité ako združovacie (pooling) vrstvy, ktorými je realizované podvzorkovanie. Výsledkom CNN je schopnosť sledovať priestorovú koreláciu [25], [42]. Na obrázku 19 je znázornená štruktúra hlbokéj neurónovej siete.



Obrázok 19 Štruktúra konvolučnej neurónovej siete [38]

3.3.4 Generatívna kompetitívna sieť

Generatívna kompetitívna sieť (GAN) je DNN, ktorá je zložená z generátora a diskriminátoru. Úlohou generátora je generovať vstupný obraz na základe náhodných vstupných dát. Úlohou diskriminátoru je rozhodovať, či je daný obraz skutočný, alebo umelo vytvorený generátorom. Cieľom učenia je, aby generátor generoval umelý obraz dostatočne verný na to, aby bol v 50% prípadoch považovaný diskriminátorom za skutočný. Pri tréningu je využívané spätné šírenie chyby [43]. Na obrázku 20 je znázornená štruktúra generatívnej kompetitívnej siete.



Obrázok 20 Štruktúra generatívnej kompetitívnej siete [44]

3.4 Využitie učenia v kompresii obrazu

Vyššie spomínané metódy je možné rozdeliť na dva druhy: kompletná (end-to-end) metóda, použiteľná ako náhrada konvenčného kodeku, alebo implementácia za účelom zefektívnenia jednotlivých častí konvenčného kodeku [25].

3.4.1 Siete určené na end-to-end spracovanie

Siete realizujúce kompletnú kompresiu sú rozdelené podľa zložitosti na:

Základnú sieť – ANN zložené zo vstupnej vrstvy, výstupnej vrstvy a skrytej vrstvy s dimenziou menšou ako dimenzia vstupnej vrstvy.

Hierarchickú sieť – DNN, ktorej jednotlivými skrytými vrstvami je znížená korelácia medzi pixelmi v obraze.

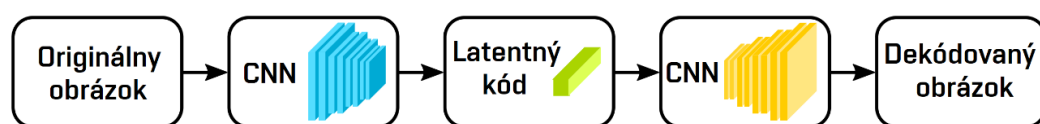
Adaptívnu sieť – sieť zložená z viacerých ANN a DNN s rôznym počtom skrytých vrstiev, realizujúcich spracovanie a kompresiu jednotlivých, rôzne zložitých častí rovnakého obrazu [25].

3.4.2 Siete zvyšujúce efektivitu konvenčných kodekov

Pomocou týchto sietí je možné napríklad optimalizovať nastavenie použitej transformácie, LPC, PCM a DPCM kódovania, kvantizačných tabuliek alebo entropického kódovania [25].

3.5 Latentná reprezentácia obrazu

Dáta spracovávané umelými a hlbokými neurónovými sieťami sú latentnými transformátormi transformované do latentnej domény a výstupom sietí sú latentné dáta (latentný kód) [25]. Na obrázku 21 je zobrazená reprezentácia dát v latentnej doméne.



Obrázok 21 Latentná reprezentácia obrazových dát [25]

Latentný kód existuje v podobe jednodimenzionálneho vektoru. V latentnom kóde sú uložené vlastnosti obrazu extrahované natrénovaným modelom, ktoré je možné využiť rôznym spôsobom. Jeden z týchto spôsobov je dekodovanie celého obrázku, alebo len jeho časti, komprimovaných pomocou ANN alebo DNN.

3.6 JPEG AI

Cieľom aktivity JPEG AI je vývoj kodekov slúžiacich na kompresiu obrazu založených na využití ML a DL, ktorými bude možné dosiahnuť vyššiu kvalitu kompresie pri porovnaní s konvenčnými kodekmi. Niektoré kodeky slúžia ako end-to-end metódy,

d'alsie ako zdokonalenia konvenčných kodekov alebo ich častí. Dataset obrázkov JPEG AI pozostáva z 3 častí: tréningového datasetu slúžiaceho na trénovanie modelu využitého na kompresiu obrázkov založenú na učení, validačného datasetu slúžiaceho na overenie konvergencie trénovaného algoritmu použitého trénovaným modelom a skrytého testovacieho datasetu slúžiaceho na vyhodnotenie výkonu vytvoreného modelu. Rozlíšenie obrázkov JPEG AI datasetu je od 256×256 pixelov až do 8K s bitovou hĺbkou 8 bitov. Cieľové kvality komprimovaných obrázkov sú uvedené v tabuľke 6, kde sú hrubým písmom vyznačené kvality (BPP) používané pri hodnotení pomocou metódy Bjøntegaard-Delta (BD-Rate) [25], [45], [46].

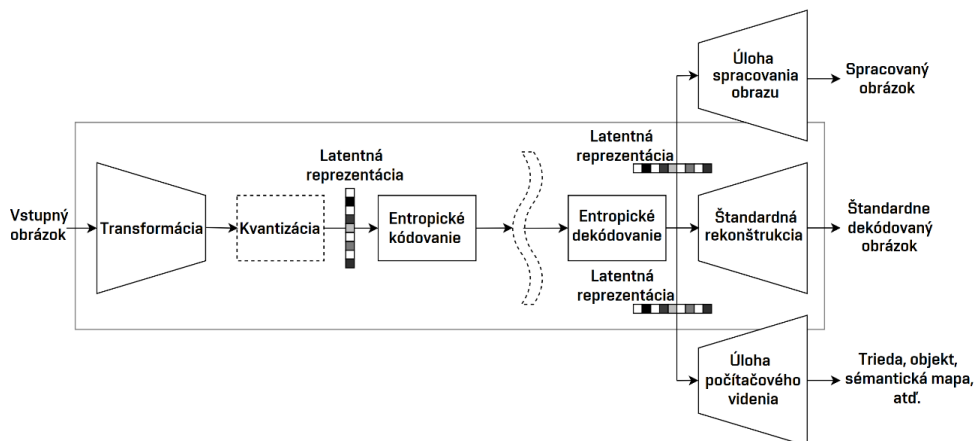
Tabuľka 6 Cieľové kvality pre kompresiu pomocou metód JPEG AI

Poradie	Bitový tok [BPP]
1	0,03
2	0,06
3	0,12
4	0,25
5	0,50
6	0,75
7	1,00
8	1,50
9	2,00

Kvalita je podľa rovnice (2) vypočítaná:

$$BPP = \frac{N_TOT_BITS}{N_TOT_PIXELS'} \quad (2)$$

kde N_TOT_BITS znamená celkový počet bitov komprimovaného obrázku a N_TOT_PIXELS' znamená celkový počet pixelov zrekonštruovaného obrázku [25], [45], [46]. Na obrázku 22 je zobrazená bloková schéma frameworku JPEG AI.



Obrázok 22 JPEG AI framework [45]

Pomocou frameworku JPEG AI je možné vykonávať okrem kompresie a dekompresie obrázkov aj ďalšie spracovanie obrazu a využiť ho pre úlohy počítačového videnia. Možné úlohy spracovania obrazu predstavujú umelé zvyšovanie rozlíšenia, redukcia šumu a rôzne vylepšenia obrázkov. Úlohy počítačového videnia sú klasifikácia, regresia, rozpoznávanie objektov alebo detekcia tváre a mnohé ďalšie [25], [45].

3.7 Kompresia planárneho a 360° obrazu založená na využití strojového a hlbokého učenia

Ako obraz v 2D časovej a priestorovej doméne, tak aj latentný kód môže byť komprimovaný. Na kompresiu sú najčastejšie používané autoenkodéry. Latentné transformátory obsahujú množstvo parametrov, ktoré boli získané učením na reálnych dátach, čím sú efektívnejšie, než konvenčné prístupy v podobe DCT alebo DWT. Cieľom je zvýšenie rozptylu bez veľkej straty kvality, čím bude umožnené skomprimovať latentný kód čo najviac. S týmto zámerom je použitý neuniformný kvantizátor, negatívne orezanie prvkov latentného kódu a výpočet kvantizačných úrovní na základe distribúcie prvkov latentného kódu. Kvantované prvky latentného kódu sú entropicky kódované. Myšlienkou tohto entropického kódovania je využitie štatistiky latentnej reprezentácie a modelu podmienenej pravdepodobnosti, ktorý je end-to-end naučený. Modely entropického kódovania sú presné a prispôsobiteľné charakteristikám kódovaného obrazu. Kodérom je generovaná kompaktná reprezentácia a úlohou dekodéra je pokúšať sa o rekonštrukciu originálneho obrazu z compactnej reprezentácie s čo najväčšou vernosťou. Zvyčajne sú tieto kodéry a dekodéry iteratívne trénované určitou funkciou, ktorou je zohľadnená kvalita rekonštruovaného obrazu a rozpätie BPP. Diferencovateľné metriky sú spojené s nastavením toku, čím je možné ďalej znížiť BPP. Nepriaznivé straty tiež môžu byť použité na zlepšenie vizuálnej kvality a poskytujú vysokú kvalitu rekonštrukcie pri nízkom BPP. Aplikácie kompresie obrazu založené na učení sú charakterizované parametrami, akými sú použitá architektúra, procesné vrstvy, typy objektívnych metrik využitých ako stratová funkcia, veľkosť procesnej jednotky, cieľová kvalita, jeden alebo viacero BPP, spôsobom nastavenia BPP a nástrojmi priestorovej korelácie. Obraz môže byť rozdelený na regióny, kde každý región je kódovaný inou kvalitou [47].

Pre kompresiu 360° obrazu je možné použiť techniky kompresie planárneho obrazu, akými sú konvolúcia, posuv, iteratívna agregácia, združovanie a premiešavanie pixelov, ale je potrebná ich adaptácia do sférickej domény, pretože skreslenie vnášané projekciou je závislé na pozícii vo sfére. Vďaka tomu je značne znížené BPP a vysokofrekvenčné detaily sú zachované vo vyššej kvalite [48].

3.7.1 Full Resolution Image Compression with Recurrent Neural Networks

Pre kompresiu algoritmom FRICwRNN [49] je použitý variačný autoenkodér s GRU

architektúrou siete s konvolučnými vrstvami a natrénovaným modelom. Tento algoritmus je trénovaný na kompresiu planárneho obrazu. Ďalšími jeho časťami sú binarizér a CNN pre entropické kódovanie. Úlohou entropického kodéra je zachytávanie závislostí medzi jednotlivými kódovými jednotkami, ktorých veľkosť je 32×32 pixelov. Výhodou variačného AE je možnosť variabilného nastavenia kompresie do viacerých kvalít.

Pri tréovaní siete sú váhy neurónov zdieľané medzi jednotlivými iteráciami a zvyškové chyby, teda rozdiely medzi pôvodným a komprimovaným obrazom, sú prenášané do ďalšej tréovacej iterácie. Ako aktivačná funkcia neurónov je použitá funkcia ELU. V rekurentných jednotkách sú zahrnuté dva konvolučné filtre, jeden na vstupe jednotky a druhý v rekurentnej časti jednotky. Kódové jednotky s rozmerom $32 \times 32 \times 3$, kde posledný rozmer reprezentuje RGB hodnoty každého pixelu, sú po kompresii zmenšené na rozmer $2 \times 2 \times 32$ v binarizovanom formáte.

Entropia binarizovaných kódov každej kódovej jednotky nie je maximálna. Pre bezstratové entropické kódovanie je použitý entropický kódér s jednou iteráciou. Pre zachytenie závislostí medzi jednotlivými binarizovanými kódmi sú použité konvolučné vrstvy a LSTM vrstva a tieto závislosti sú medzi sebou kódované progresívne.

3.7.2 Context-adaptive Entropy Model for End-to-end Optimized Image Compression

Pre kompresiu algoritmom CAEMfE2EOIC [50] je použitý model natrénovaného CNN autoenkodéru, využívajúci adaptívne entropické kódovanie na základe kontextu, čím je dosiahnutá vysoká presnosť odhadu distribúcie latentného kódu. Tieto kontexty sú kontext spotrebovania bitu a kontext uvoľnenia bitu. Vďaka týmto kontextom je možný presnejší odhad distribúcie latentných reprezentácií s použitím obecnějších entropických modelov a efektívnejšia redukcia priestorových závislostí susediacich latentných reprezentácií.

Na strane kodéru je obraz normalizovaný, transformovaný do latentnej reprezentácie, kvantovaný zaokrúhlením a kódovaný natrénovaným entropickým kodérom. V dekódovacej časti je latentný kód entropicky dekódovaný rovnakým entropickým modelom, aký bol použitý pri kódovaní, následne je dekódovaný latentný kód inverzne transformovaný a denormalizovaný.

Veľkosť kódovej jednotky je 256×256 pixelov. Aktivačné funkcie neurónov sú ReLU a Leaky ReLU. Metriky použité pre tréovanie modelov sú MSE a MS-SSIM a algoritmus je trénovaný na kompresiu planárneho obrazu. Celkový počet natrénovaných modelov je osemnásť, čo reprezentuje deväť cieľových obrazových kvalít v oboch použitých metrikách.

3.7.3 High-Fidelity Generative Image Compression

Pre kompresiu algoritmom HiFiC [51] je použitá GAN sieť s natrénovaným modelom. S jej využitím je možné dosiahnuť viac ako 2-násobnú úsporu BPP v porovnaní

so staršími riešeniami využívajúcimi učenie.

Architektúra siete je zložená z kodéra obrazu, modelu pravdepodobnosti, generátoru a diskriminátoru. Kodér, generátor a diskriminátor sú tvorené konvolučnými a normalizačnými vrstvami. V kodéri je obraz rozložený na kódové jednotky s veľkosťou 256×256 pixelov, stratovo zakódovaný do latentnej reprezentácie a kvantovaný zaokrúhlením. Pre následné bezstratové uloženie latentného kódu je použitý model pravdepodobnosti s aritmetickým entropickým kodérom. V generátore je rekonštruovaný obraz vytvorený dekodovaním latentného kódu. Aktivačná funkcia neurónov kodéru a generátoru je ReLU. Vstupy diskriminátoru sú tvorené výstupom z kvantizéru a výstupom z generátoru. Cieľom je natrénovať generátor tak, aby generátorom rekonštruovaný obraz bol považovaný diskriminátorom za skutočný. Aktivačná funkcia neurónov diskriminátoru je Leaky ReLU a aktivačná funkcia neurónov poslednej vrstvy diskriminátoru je sigmoidálna.

Metriky použité pre trénovanie modelov sú MSE a LPIPS, ktoré sú použité naraz. Model HiFiC-low je natrénovaný pre cieľové BPP s hodnotou 0,14, model HiFiC-med je natrénovaný pre BPP s hodnotou 0,30 a model HiFiC-high je natrénovaný pre cieľové BPP s hodnotou 0,45. Algoritmus je trénovaný na kompresiu planárneho obrazu.

3.7.4 End-to-End Optimized 360° Image Compression

Pretože 360° obraz kódovaný pomocou ERP projekcie je regionálne redundantne nadvzorkovaný, je vhodné jednotlivé regióny adaptívne podvzorkovať, čím je riadená alokácia častí latentného kódu prislúchajúcich daným regiónom. Pre kompresiu algoritmom LIC360 je použitá DNN architektúra optimalizovaná pre kompresiu 360° obrázkov v ERP projekcii [52]. Veľkosť kódovej jednotky je 256×256 pixelov.

Architektúra algoritmu je zložená z kodéru, dekodéru, siete s adaptívnym škálovaním zemepisnej šírky v obraze, siete odhadujúcej obrazový obsah a siete odhadujúcej entropiu. Tieto siete spolu s kodérom a dekodérom sú zložené z blokov realizujúcich podvzorkovanie, nadvzorkovanie, reziduálnych blokov a blokov zisťujúcich oblasť pozornosti. Tieto bloky pozostávajú hlavne z konvolučných vrstiev. Kodér je zložený z blokov realizujúcich podvzorkovanie, bloku zisťujúceho pozornosť v obraze a reziduálnych blokov. Výstupný kód je kvantovaný. Na určenie dôležitosti jednotlivých regiónov obrazu je vytvorená štruktúrna mapa kódu, ktorou je kvantovaný latentný kód váhovaný. Mapa je vytvorená sieťou s adaptívnym škálovaním zemepisnej šírky v obraze a sieťou odhadujúcou obrazový obsah. Váhovaný latentný kód je entropicky kódovaný sieťou odhadujúcou entropiu. Tá je odhadovaná na základe rozdelenia pravdepodobnosti latentných kódov zmesou Gaussovských rozdelení, kde je odhadovaná stredná hodnota, rozptyl a váhy pravdepodobnosti. Dekodér je zložený z blokov realizujúcich nadvzorkovanie, bloku zisťujúceho pozornosť v obraze a reziduálnych blokov. Aktivačná funkcia neurónov konvolučných vrstiev je ReLU a aktivačná funkcia poslednej vrstvy neurónov kodéru a siete odhadujúcej obrazový obsah je sigmoidálna.

Jediné možné rozlíšenie komprimovaných obrázkov je 1024×512 pixelov. Takéto nízke rozlíšenie je zvolené preto, že cieľom je kódovať výrez v danom 360° obrázku, čiže je dostatočné. Metriky použité pre tréovanie modelov sú VMSE a VSSIM, čo sú implementácie metrík MSE a SSIM s uvažovaním oblasti záujmu v 360° obraze.

4. OBJEKTÍVNE A SUBJEKTÍVNE METRIKY KVALITY

Pri sledovaní obrazu je dôležitým parametrom kvalita. Pri stratovej kompresii obrazu dochádza k zníženiu kvality obrazu. Zároveň, pri použití kompresných metód založených na ML a DL dochádza k odlišným kompresným artefaktom v obraze, ako pri použití konvenčných metód a tieto artefakty majú vysoký vplyv na vnímanú kvalitu. Na meranie kvality obrazu sú používané metriky kvality. Tie je možné rozdeliť na objektívne a subjektívne metriky. Pretože 360° má veľký zorný uhol, človek nie je pri jeho pozorovaní schopný vnímať všetky jeho časti v jednom momente. Častokrát je pozornosť zameraná na stred obrazu viac, ako na jeho okraje, pretože zo stredu je vnímaných najviac informácií. Preto je pri hodnotení kvality 360° obrazu dôležité rátať s touto vlastnosťou ľudského zraku (HVS) [1], [9].

4.1 Objektívne metriky kvality

Pomocou objektívnych metrík je kvalita obrazu určená matematicky na základe výpočtov za použitia rôznych algoritmov. Komplexnejšie objektívne metriky zohľadňujú vnímanie obrazu ľudským zrakom, v ktorom sa nachádza mnoho nedokonalostí. Všetky popísané objektívne metriky sú plne referenčné (FR), čo znamená, že pre vyhodnotenie kvality je vypočítaný rozdiel medzi originálnym a skresleným obrazom. Existujú aj čiastočne referenčné metriky (RR) využívajúce niektoré parametre pôvodného obrazu a bezreferenčné metriky (NR), ktorými je vypočítaná kvalita len na základe samotného skresleného obrazu [1].

4.1.1 Peak Signal to Noise Ratio

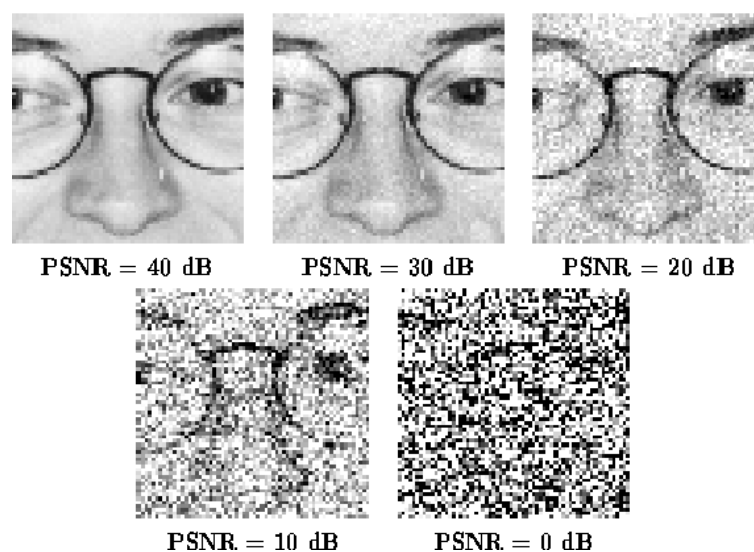
Objektívna metrika PSNR je jedna z prvých metrík používaných na hodnotenie kvality obrazu. Je v nej počítaný odstup signálu od šumu. Kvôli vysokému dynamickému rozsahu signálov v obraze je vyjadrená v logaritmickej miere v dB. Definuje sa stredná kvadratická odchýlka (MSE). Hodnota MSE je definovaná rovnicou (3):

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2. \quad (3)$$

Hodnota PSNR je definovaná rovnicou (4):

$$PSNR = 10 \log \left(\frac{MAX_i^2}{MSE} \right) [dB], \quad (4)$$

kde parameter MAX_i predstavuje maximálnu hodnotu pixelu obrazu. Pre farebný obraz je MSE delená hodnotou 3, pre každú z RGB zložiek obrazu. Typická hodnota PSNR pre komprimovaný obraz je na úrovni 30 – 40 dB [1], [9]. Na obrázku 23 možno vidieť príklad pre rôzne hodnoty PSNR.



Obrázok 23 Príklad rôznych hodnôt PSNR [53]

Metrika je jednoduchá a rýchla na výpočet. Nevýhodou tejto metriky je fakt, že nepočíta s HVS a je určená len pre klasický plochý obraz. PSNR metriky zohľadňujúce HVS sú PSNR-HVS [54], PSNR-HVS-M [55] a VSNR [56]. Metrika PSNR-HVS je rozšírením pôvodnej metriky PSNR o vnímanie kontrastu [51]. PSNR-HVS-M je rozšírením metriky PSNR-HVS o vizuálne maskovanie [47], [55]. PSNR metriky vytvorené na hodnotenie kvality 360° obrazu sú S-PSNR, WS-PSNR a CPP-PSNR [9], [11], [57]. V metrike S-PSNR je obmedzený počet vzorkovacích bodov rovnomerne rozmiestnených na sférickom povrchu, ktoré sú znovu premietnuté na originálny a skreslený obraz s cieľom nájdania zodpovedajúcich pixelov, nasledované výpočtom PSNR [9], [11], [57]. Metrika WS-PSNR je rozšírením S-PSNR, kde je škálovací faktor plochy projekcie z 2D do sféry použitý ako váhový faktor pri výpočte PSNR [9], [11], [57]. V metrike CPP-PSNR sú pixely originálneho a skresleného obrazu projektované do sférickej domény a premapované do CPP projekcie bez zmeny priestorového rozlíšenia, nasledované výpočtom PSNR [9], [11], [57].

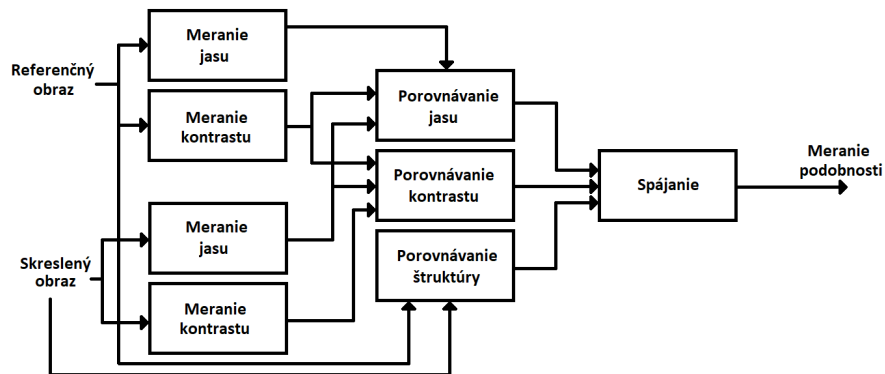
4.1.2 Structural similarity index

Objektívna metrika SSIM [58] je druhá veľmi často používaná metrika na hodnotenie kvality obrazu. Vyjadruje podobnosť dvoch obrazov. Sú v nej zohľadnené princípy HVS, preto výsledky lepšie korelujú so subjektívnym vnemom. Na výpočet sú porovnávané štrukturálne skreslenia obrazu, vyhodnotenie vizuálneho dopadu posunu jasov v obraze a zmeny kontrastu oproti pôvodnému obrazu. Pri farebnom obraze sa počíta obvykle len s jasovou zložkou signálu. Výsledkom je SSIM index, ktorý nadobúda hodnoty v rozmedzí 0 – 1. Hodnotou 1 je vyjadrená zhodnosť s pôvodným obrazom, hodnotou 0 je vyjadrená nulová podobnosť s pôvodným obrazom. Hodnota SSIM indexu je definovaná rovnicami (5) a (6):

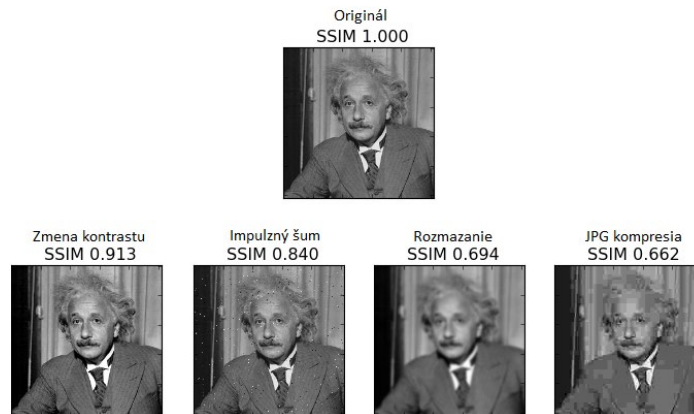
$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (5)$$

$$C_i = (K_iL)^2, \quad (6)$$

kde parametrom L je vyjadrený dynamický rozsah hodnôt pixelov a konštanta $K \ll 1$. Parametrom μ je vyjadrený vážený priemer a parametrom σ je vyjadrený vážený rozptyl [58]. Na obrázku 24 je zobrazená schéma metriky SSIM a na obrázku 25 príklad SSIM indexu pri rôznych skresleniach obrazu.



Obrázok 24 Schéma metriky SSIM [58]



Obrázok 25 Príklad rôznych skreslení obrazu a hodnôt SSIM indexu [59]

MS-SSIM [60] je vylepšením pôvodnej metriky SSIM. Vykonáva sa vo viacerých mierkach prostredníctvom procesu viacerých stupňov čiastočného vzorkovania, pripomínajúce viacúrovňové spracovanie. Dosahuje presnejšie výsledky ako metrika SSIM [47], [60].

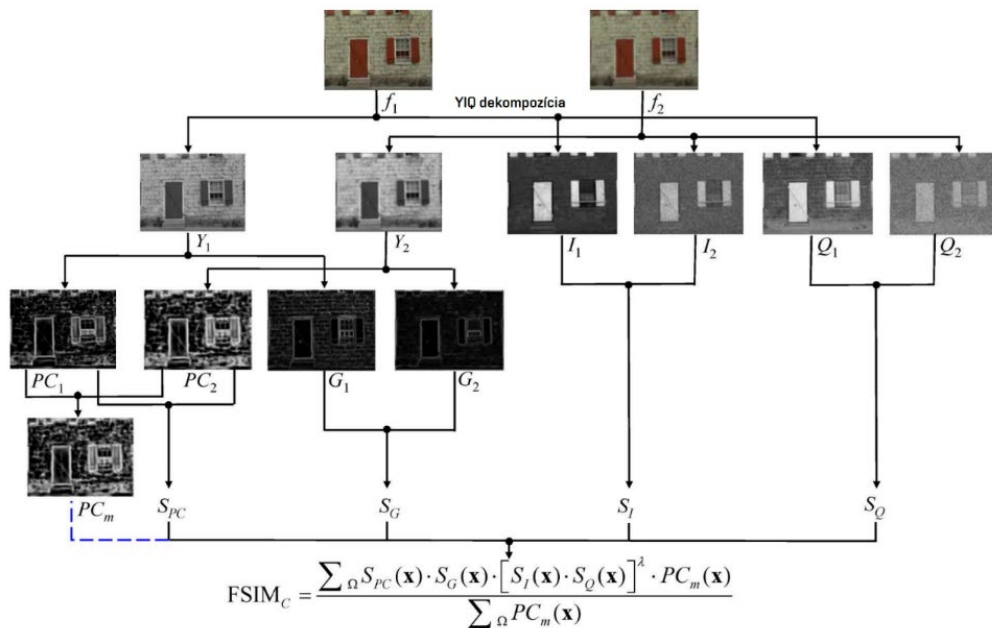
4.1.3 Visual information fidelity

Základom metriky VIFp [61] je vyhodnocovanie štatistík prírodnej scény s informáciami extrahovanými HVS a model skreslenia na výpočet informácie zdieľanej medzi

originálnym a skresleným obrazom. Pre výpočet sa obrazový signál prevádza do frekvenčnej domény pomocou DWT, v ktorej sa uskutočňuje výpočet. Dosahuje vysokú koreláciu s HVS. Výsledkom metriky je VIFp index určujúci pomer medzi originálnym a skresleným obrazom, ktorý označuje spoločnú informáciu medzi oboma obrazmi. VIFp index bežne nadobúda hodnoty 0 – 1. Hodnota indexu 1 znamená, že oba obrázky sú zhodné. Hodnota indexu menšia ako 1 znamená, že vernosť skresleného obrazu je menšia ako vernosť pôvodného obrazu. Môže sa stať, že hodnota indexu je väčšia ako 1, čo znamená, že vernosť skresleného obrazu je väčšia ako vernosť pôvodného obrazu, čo je možné dosiahnuť zvýšením kontrastu skresleného obrazu [47], [61].

4.1.4 Feature similarity

V metrike FSIM [62] sú na výpočet kvality využívané nízkoúrovňové vlastnosti HVS, akou je fázová zhoda, ktorou je charakterizovaná významnosť lokálnych štruktúr a ako druhý parameter je použitá magnitúda obrazového gradientu. FSIMc je verziou metriky FSIM rozšírenou o výpočet kvality zahrňujúci chrominančné signály obrazu. Výsledky metriky sú vo vysokej korelácii s HVS a nadobúdajú hodnoty v rozmedzí 0 – 1, kde vyššia hodnota znamená vyššiu vizuálnu kvalitu pri porovnaní originálneho a skresleného obrazu [47], [62], [63]. Schéma metriky FSIMc je zobrazená na obrázku 26.

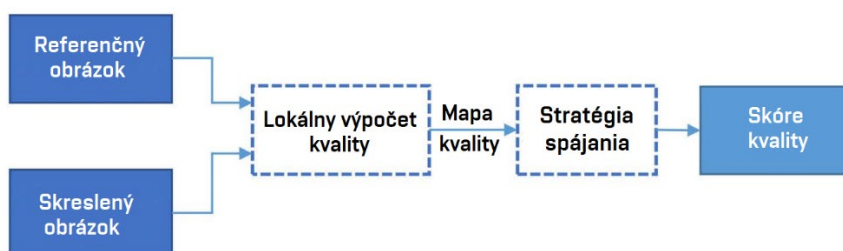


Obrázok 26 Schéma metriky FSIMc [62]

4.1.5 Gradient magnitude similarity deviation

V metrike GMSD [64] je základom vytvorenie lokálnej mapy kvality. HVS je veľmi citlivý na gradienty obrazu. Gradienty obrazu sú citlivé na skreslenie obrazu a lokálne štruktúry v skreslenom obraze trpia rôznym stupňom degradácie. Pixelová gradientová

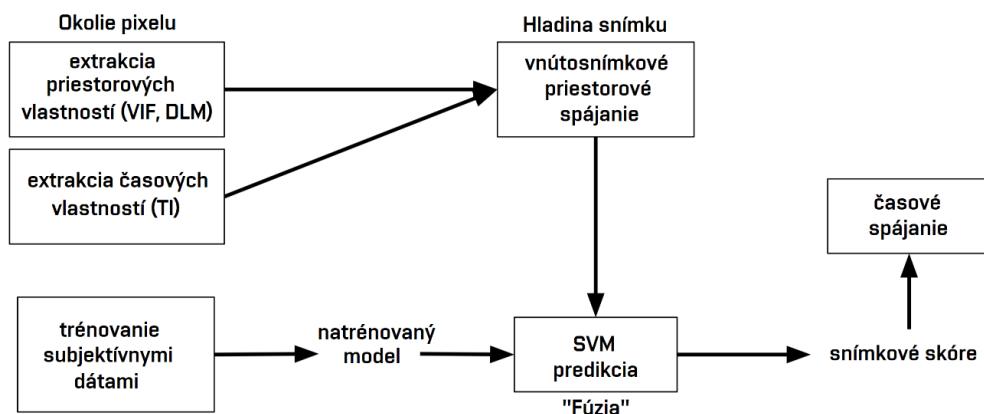
podobnosť (GMS) medzi originálnym a skresleným obrazom v kombinácii so štandardnou odchýlkou GMS mapy dokáže presne predpovedať kvalitu obrazu. Výpočtový algoritmus GMSD je rýchlejší ako väčšina najmodernejších metód hodnotenia kvality obrazu a poskytované výsledky sú veľmi presné. Výsledkom je GMSD index, ktorého nižšia hodnota znamená vyššiu kvalitu a vyššiu podobnosť skresleného a originálneho obrazu. Schéma metriky GMSD je zobrazená na obrázku 27.



Obrázok 27 Schéma metriky GMSD [64]

4.1.6 Video multi-method assessment fusion

V metrike VMAF [65] je odhadovaná subjektívna kvalita na základe rozdielov medzi originálnym a skresleným obrazom. Na výpočet je využité strojové učenie a SVM regresiou sú výsledky viacerých metrických zlúčených do jedného výsledku. Medzi tieto metriky patria VIFp, DLM, MCPD a AN-SNR. Tieto metriky využívajú parametre HVS, akými sú SNR, vizuálna vernosť, poruchy rušiacie pozornosť, strata detailov a časové rozdiely v jasovej zložke obrazu. Na výpočet kvality sú používané modely pre rôzne rozlíšenia obrazu. Tieto modely sú vytvárané subjektívnym hodnotením kvality skupinou pozorovateľov. Výsledkom je VMAF index s hodnotami v rozmedzí 0 – 100. Hodnota 100 predstavuje kvalitu identickú s pôvodným obrazom. Pre dosiahnutie správnych výsledkov musí mať originálny a skreslený obraz rovnaké rozlíšenie. Výsledky dosiahnuté touto metrikou vysoko korelujú s HVS a je vhodná pre použitie na 360° obrázky a video [47], [65], [66]. Na obrázku 28 je zobrazená schéma metriky VMAF.



Obrázok 28 Schéma metriky VMAF [65]

4.1.7 Normalized Laplacian pyramid distance

V metrike NLPD [67] je vypočítaná kvalita na základe HVS vlastností, ktorými sú lokálny jasový rozdiel a kontrola lokálneho zisku. Obraz je dekomponovaný použitím Laplaceovej pyramídy, ktorou je rozdelený lokálny odhad stredného jasu vo viacerých mierkach. Skóre metriky je počítané v normalizovanej laplaceovej doméne, kde je relatívna kvalita skresleného obrazu odmocninou chyby k referenčnému obrazu v normálovo váhovanej laplaceovej doméne. Nižšie výsledné skóre metriky znamená vyššiu kvalitu skresleného obrazu a výsledky tejto metriky sú presnejšie ako výsledky získané metrikou MS-SSIM [47], [67].

4.2 Subjektívne metriky kvality

Pri subjektívnych metrikách je kvalita obrazu hodnotená ľudskými respondentmi na základe priameho pozorovania obrazu. Existuje viacero prístupov, ako subjektívne hodnotiť kvalitu obrazu. V základe je potrebné mať určitý minimálny počet respondentov, ktorý je podľa odporúčaní Rec. ITU-R BT.500-14 [47], [68] a ITU-T P.910 [47], [69] určený na 15, aby bolo možné považovať dosiahnuté výsledky za relevantné. Vo výnimočných prípadoch je možné namiesto 15 bežných respondentov hodnotiť kvalitu 4 – 8 respondentmi, ktorí sú odborníkmi, alebo inými kritickými subjektmi v danej oblasti. Čím je počet respondentov hodnotiacich kvalitu vyšší, tým presnejšie sú dosiahnuté výsledky.

4.2.1 Double stimulus impairment scale

Hodnotenie kvality metrikou DSIS [68] spočíva v premietnutí 2 obrazov respondentovi za sebou s krátkou prestávkou. Prvý obraz je referenčný a druhý je skreslený. Pri prvej variante je referenčný obraz zobrazený po dobu 10 sekúnd, nasleduje 3-sekundová prestávka a následne je zobrazený skreslený obraz po dobu 10 sekúnd. Respondent hodnotí kvalitu od momentu zobrazenia skresleného obrazu a má čas ďalších 5 – 11 sekúnd po skončení zobrazenia na hodnotenie kvality. Pri druhej metóde je referenčný obraz zobrazený po dobu 10 sekúnd, nasleduje 3-sekundová prestávka, nasleduje skreslený obraz zobrazený po dobu 10 sekúnd, nasleduje 3-sekundová prestávka, nasleduje referenčný obraz zobrazený po dobu 10 sekúnd, nasleduje 3-sekundová prestávka, nasleduje skreslený obraz zobrazený po dobu 10 sekúnd a nakoniec dodatočných 5 – 11 sekúnd na hodnotenie kvality, kde je kvalita hodnotená od druhého zobrazenia referenčného obrazu. Hodnotenie kvality nadobúda hodnoty v rozmedzí 1 – 5, ktoré je bližšie popísane v tabuľke 7 [68].

Tabuľka 7 Hodnotenie kvality metódou DSIS

	1	2	3	4	5
Hodnotenie	Veľmi obťažujúce	Obťažujúce	Mierne obťažujúce	Pozorovateľné, neobťažujúce	Neobťažujúce

4.2.2 Double stimulus continuous quality scale

Metóda DSCQS [68] je podobná metóde DSIS. Hlavným rozdielom je nevedomosť respondenta v tom, ktorý z obrazov je referenčný a ktorý je skreslený. Stupnica hodnotenia je rovnaká ako v prípade DSIS, ale rozdiel v spôsobe hodnotenia je normovanie stupnice na hodnoty 0 – 100, kde 0 znamená zlú kvalitu a 100 výbornú kvalitu [47], [68].

4.2.3 Absolute category rating

Pri metrike ACR [69] je respondentovi zobrazovaný skreslený obraz po dobu 10 sekúnd, za ktorým nasleduje prestávka s trvaním 10 sekúnd na hodnotenie kvality. Jednotlivé obrázky s prestávkami sú zobrazované kontinuálne za sebou. Na hodnotenie kvality sa používa stupnica 5 slovných známkok, ktorá je zobrazená v tabuľke 8.

Tabuľka 8 Hodnotenie kvality metrikou ACR

	1	2	3	4	5
Hodnotenie	Zlá kvalita	Nízka kvalita	Prijateľná kvalita	Dobrá kvalita	Výborná kvalita

4.2.4 Absolute category rating with hidden reference

Metrika ACR-HR [69] je rozdielna od metriky ACR tým, že je medzi jednotlivými skreslenými obrazmi zobrazený aj referenčný obraz a respondent nevie, ktorý z obrazov je referenčný a ktorý je skreslený, zobrazovanie jednotlivých obrazov je náhodné. Hodnotenie kvality referenčného obrazu by malo byť najvyššie. Ak to tak nie je, respondent sa dopúšťa chyby testovania. Stupnica hodnotenia je rovnaká ako pri metrike ACR. Spracovanie výsledkov prebieha odčítaním známky referenčného obrazu $V(REF)$ od známky skresleného obrazu $V(PVS)$ a pripočítaním hodnoty 5 a rovnica (7) je definovaná ako:

$$DV(PVS) = V(PVS) - V(REF) + 5, \quad (7)$$

kde $DV(PVS)$ označuje diferenčné skóre pozorovateľa. Ak by respondent ohodnotil skreslený obraz ako kvalitnejší než je referenčný, platí rovnica (8):

$$crushed_DV = (7 \times DV) / (2 + DV), \quad (8)$$

kde $crushed_DV$ je znižovacia funkcia [69].

5. PRAKTICKÁ ČASŤ

Hlavnou časťou tejto práce je vytvorenie frameworku, ktorým bude možné komprimovať 360° snímky s využitím algoritmov hlbokého učenia. Tvorba datasetu, návrh frameworku, spracovanie a kompresia obrazu a výpočet metrík sú uskutočnené na notebooku ASUS ROG Strix SCAR III G731GW-EV042T, ktorého parametre sú zhrnuté v tabuľke 9.

Tabuľka 9 Hardwarové a softwarové špecifikácie použitého notebooku

Komponent	Názov komponentu	Špecifikácie a vlastnosti
CPU	Intel Core i7-9750H	6 jadier, 12 vlákien, 2,6 – 4,5 GHz
GPU 1	Intel UHD Graphics 630	Pamäť zdieľaná z RAM
GPU 2	NVIDIA GeForce RTX 2070	8 GB, CUDA, TensorRT
RAM	Kingston HyperX DDR4	32 GB, 2666 MHz, CL 15
SSD 1	Intel 660p	512 GB, QLC, stránkovací súbor
SSD 2	Crucial MX500	2 TB, TLC
OS	Windows 10	Home Edition 22H2 x64
PSU	ASUS	230 W

Z tabuľky 9 vidno, že aj keď sa v notebooku nenachádzajú najnovšie a najvýkonnejšie komponenty (3 – 4 roky staré), sú na dnešné pomery stále dosť výkonné. Procesor (CPU), ktorým je možné dosiahnuť uspokojivé rýchlosti výpočtu obsahuje SIMD a AVX inštrukčné sady pre spracovanie viacero dát jednou inštrukciou, ako aj pokročilé vektorové rozšírenia. Vďaka tomu je možné výpočty značne urýchliť v porovnaní s obyčajnými skalárnymi inštrukciami. Avšak pri ML a hlavne DL je možné výpočty urýchliť ešte viac pomocou grafickej karty (GPU), pretože optimalizované ML a DL algoritmy je možné vykonávať prostredníctvom paralelných operácií. Na to je vhodná grafická karta NVIDIA, pretože väčšina algoritmov počítaných pomocou GPU je vytvorených pre platformu CUDA, ktorá je vytvorená pre GPU NVIDIA. Vybrané GPU NVIDIA posledných 3 generácií navyše obsahujú špeciálne Tensor jadrá, ktoré sú hardwarovo vytvorené pre ML a DL operácie, čím je možné celý proces výpočtu ďalej značne zrýchliť. Okrem dostatočne výkonného HW je potrebné použiť aj vhodný SW. Niektoré nástroje boli dostupné len vo forme zdrojového kódu, takže tieto zdrojové kódy boli ďalej spracované nástrojom Cmake [70] vo verzii 3.25.0 x64 a skompilované v Microsoft Visual Studio 2022 [71] vo verzii 17.4.1 x64.

5.1 Teoretický návrh frameworku

Na vytvorenie frameworku je potrebné najskôr vytvoriť dataset vhodných 360° obrázkov. Druhým krokom je skomprimovanie týchto obrázkov pomocou konvenčných kodekov a kodekov založených na využití ML a DL. Aby mohli byť tieto obrázky komprimované kodekmi založenými na ML a DL, je potrebné najskôr natrénovať potrebné modely. Použité implementácie DL algoritmov už obsahujú natrénované modely. Po kompresii a dekompresii obrázkov je ďalším krokom vyhodnotenie kvality týchto obrázkov. Frameworkom bude možné komprimovať 360° snímky, kde odporúčaná projekcia je ERP, pretože väčšina obrázkov dostupných vo verejných internetových databázach je zakódovaných práve v ERP projekcii. Kodeky dostupné vo frameworku budú JPEG, HEIC, AVIF, JPEG XL, VVC Intra a algoritmy založené na využití ML a DL: FRICwRNN, CAEMfE2EOIC, HiFiC a LIC360. Odporúčané rozmery vstupných obrázkov sú v násobkoch 256 pre každú dimenziu 2D priestoru obrázku, maximálne odporúčané rozlíšenie je do veľkosti 8192×4096 pixelov pre akceptovateľnú výpočtovú náročnosť. Zároveň je toto vysoké rozlíšenie potrebné pre dostatočne vysokú kvalitu reprodukcie, pretože pri sledovaní 360° obrazu vo VR prostredí je zobrazovaný len výrez z celkového obrazu [1] a rozlíšenie obrazu z tohto výrezu by pre zachovanie vizuálnej kvality malo byť rovné rozlíšeniu VR headsetu, alebo vyššie. Výstupom frameworku budú obrázky vo formátoch jednotlivých kodekov s používateľom nastavenou kvalitou kompresie a dekomprimované verzie obrázkov vo formáte PNG. Pre určenie kvality kompresie budú frameworkom počítané hodnoty metrik PSNR-HVS-M, WS-PSNR, MS-SSIM, VIFp, FSIMc, GMSD a VMAF. Frameworkom bude zaznamenaný a zobrazený čas kompresie pomocou vybraných kodekov. Dekomprimované obrázky v PNG formáte je možné jednoducho zobrazovať a posúdiť kvalitu kompresie aj subjektívne. Framework bude naprogramovaný v jazyku Python 3.9. Praktická realizácia frameworku je popísaná v kapitole 7.

5.2 Dataset obrázkov

Dataset obrázkov je zložený z 10 obrázkov s 360° uhlom záberu. Tieto obrázky sú získané z verejne dostupných databáz obrázkov a videosekvencií KandaoVR [72] a ESO [73] a ich vlastnosťou sú rozličné priestorové indexy. Obrázky z videosekvencií boli získané extrahovaním snímkov z jednotlivých videosekvencií využitím programu pre prehrávanie médií MPC-HC [74] vo verzii 1.9.24 x64. Kategórie obrázkov sú: ľudia, zvieratá, príroda, panoráma, nočná obloha, ulica, mesto, interiér a exteriér. Všetky obrázky sú v rozlíšení 8192×4096 pixelov a ďalej boli podľa potreby zmenšené aj na rozmery 1024×512, 2048×1024 a 4096×2048, čiže celkovo sú použité 4 rôzne rozmery. Zmenšenie bolo vykonané voľným programom na prezeranie a jednoduché spracovanie obrázkov FastStone Image Viewer [75] vo verzii 7.7. Náhl'ady obrázkov sú v prílohe A.

5.3 Priestorový index obrázkov

Priestorový index (SI) je parameter, ktorým je určené množstvo priestorových detailov v obraze [68], [69]. V podstate je touto hodnotou vyjadrená komplexnosť scény. Vyšší priestorový index predstavuje vyššie množstvo detailov a tým aj vyššiu komplexnosť scény. SI je vypočítaný aplikovaním Sobelovho filtru na lumenančnú zložku obrazu a následne je z takto filtrovaného obrazu vypočítaná smerodajná odchýlka jednotlivých obrazových bodov.

Na výpočet SI bol použitý program SITI [76], ktorým je možné okrem SI merať aj časový index (TI). TI parameter je používaný v prípade videosekvencií. Jednotlivé parametre SITI sú zobrazené v tabuľke 10. Príkaz na výpočet SI je nasledovný:

```
siti --input-file <názov>.png
```

Tabuľka 10 Parametre programu SITI

Parameter	Funkcia parametru
siti	Spustenie programu SITI
--input file	Vstupný súbor
<názov>.png	Názov súboru, súbor vo formáte PNG

Výsledky SI pre jednotlivé obrázky sú uvedené v prílohe A pod náhľadmi týchto obrázkov. Tieto obrázky sú pôvodné obrázky v rozlíšení 8K.

5.4 Spracovanie, kompresia a dekompresia konvenčnými kodekmi

V nasledujúcich podkapitolách je uvedený celý postup spracovania obrázkov od prípravy a konverzie formátov až po kompresiu a dekompresiu. Všetky použité implementácie konvenčných kodekov sú v ich najaktuálnejších verziách.

5.4.1 Konverzia formátov obrázkov nástrojom FFmpeg

Obrázky nachádzajúce sa v datasete sú uložené v bezstratovom formáte PNG. Vstupným formátom obrázkov algoritmov na báze ML a DL je tiež formát PNG. Rôznymi kodekmi, nástrojmi a objektívnymi metrikami sú vyžadované rôzne formáty vstupných obrázkov uvedené aj v tabuľke 11, preto je potrebné tieto formáty prekonvertovať. Konverzia je vykonaná voľným open-source nástrojom FFmpeg [77] vo verzii 5.1.2 x64. Jednotlivé parametre pre konverziu formátov sú vysvetlené v tabuľke 12. Pre kontrolu vytváraných YUV a Y4M súborov bol použitý program YUView [78] vo verzii 2.13. Kompresia jednotlivými kodekmi bola použitá vždy s najvyšším nastavením efektivity daného kodéru, pretože aj za cenu dlhšieho času kompresie bol tento čas stále akceptovateľný.

Tabuľka 11 Vyžadované formáty obrázkov jednotlivých kodekov, nástrojov a metrick

Kodek/nástroj/metrika	Vstupný formát	Výstupný formát
JPEG	BMP	JPG/BMP
HEIC	YUV	BIN/YUV
AVIF	PNG	AVIF/PNG
JPEG XL	PNG	JXL/PNG
VVC Intra	YUV	266/YUV 420p10LE
VQMT	YUV	-
360tools_metric	YUV	-
FSIM	BMP/PNG	-
GMSD	PGM	-
VMAF	Y4M	-
TensorFlow	PNG	PNG
PyTorch	PNG	PNG

Tabuľka 12 Parametre konverzie formátov

Parameter	Funkcia parametru
-i	Vstupný súbor
-pix_fmt	Nastavenie farebného formátu
rgb24	Formát RGB, 8 bitov na kanál
yuv420p	Formát YUV, podvzorkovanie 4:2:0, 8 bitov na kanál
yuv420p10le	Formát YUV, podvzorkovanie 4:2:0, 10 bitov na kanál, Little Endian
yuv444p	Formát YUV, podvzorkovanie 4:4:4, 8 bitov na kanál
-video_size	Rozlíšenie obrazu
W	Horizontálny rozmer v pixeloch (width)
H	Vertikálny rozmer v pixeloch (height)
<názov>	Názov súboru

Konverzia z formátu BMP do formátov PNG, YUV444, YUV420, Y4M(444) a PGM:

```
ffmpeg -i "<názov>.bmp" -pix_fmt rgb24 "<názov>.png"
ffmpeg -i "<názov>.bmp" -pix_fmt yuv444p "<názov>.y4m"
ffmpeg -i "<názov>.bmp" -pix_fmt yuv444p "<názov>.yuv"
ffmpeg -i "<názov>.bmp" -pix_fmt yuv420p "<názov>.yuv"
ffmpeg -i "<názov>.bmp" "<názov>.pgm"
```

Konverzia z formátu PNG do formátov BMP, YUV444, YUV420, Y4M(444) a PGM:

```
ffmpeg -i "<názov>.png" -pix_fmt rgb24 "<názov>.bmp"
ffmpeg -i "<názov>.png" -pix_fmt yuv444p "<názov>.y4m"
ffmpeg -i "<názov>.png" -pix_fmt yuv444p "<názov>.yuv"
ffmpeg -i "<názov>.png" -pix_fmt yuv420p "<názov>.yuv"
ffmpeg -i "<názov>.png" "<názov>.pgm"
```


Konverzia z formátu YUV420 do formátov YUV444 a Y4M(444):

```
ffmpeg -video_size WxH -pix_fmt yuv420p -i "<názov>.yuv" -pix_fmt yuv444p "<názov>.yuv"
ffmpeg -video_size WxH -pix_fmt yuv420p -i "<názov>.yuv" -pix_fmt yuv444p "<názov>.y4m"
```

Konverzia z formátu YUV420p10LE do formátov YUV420, YUV444 a Y4M(444):

```
ffmpeg -video_size WxH -pix_fmt yuv420p10le -i <názov>.yuv -pix_fmt yuv420p <názov>.yuv
ffmpeg -video_size WxH -pix_fmt yuv420p10le -i <názov>.yuv -pix_fmt yuv444p <názov>.yuv
ffmpeg -video_size WxH -pix_fmt yuv420p10le -i <názov>.yuv -pix_fmt yuv420p <názov>.y4m
```

Konverzia z formátu YUV444 do formátu Y4M(444):

```
ffmpeg -video_size WxH -pix_fmt yuv444p -i "<názov>.yuv" "<názov>.y4m"
```

Konverzia z formátu Y4M do formátu PNG:

```
ffmpeg -i "<názov>.y4m" "<názov>.png"
```

5.4.2 Kompresia kodekom JPEG

Komprimovanie a dekomprimovanie obrázkov kodekom JPEG bolo uskutočnené referenčným SW jpeglib-turbo [79] vo verzii 2.1.4 x64. V tabuľke 13 sú zobrazené parametre kodéru a v tabuľke 14 parametre dekodéru JPEG.

Tabuľka 13 Parametre kodéru JPEG

Parameter	Funkcia parametru
cjpeg	Spustenie kodéru JPEG
-quality	Nastavenie kvality, 0 (najnižšia) – 100 (najvyššia)
-optimize	Optimalizácia Huffmanovej tabuľky
-outfile	Výstupný/komprimovaný súbor
<názov>.bmp	Vstupný/referenčný súbor

Tabuľka 14 Parametre dekodéru JPEG

Parameter	Funkcia parametru
djpeg	Spustenie dekodéru JPEG
-bmp	Formát výstupného súboru BMP
-outfile	Výstupný/dekomprimovaný súbor
<názov>.jpg	Vstupný/komprimovaný súbor

Kompresia a dekompresia kodekom JPEG:

```
cjpeg -quality <číslo> -optimize -outfile <názov>.jpg <názov>.bmp  
djpeg -bmp -outfile <názov>.bmp <názov>.jpg
```

5.4.3 Kompresia kodekom HEIC

Komprimovanie a dekomprimovanie obrázkov kodekom HEIC bolo uskutočnené referenčným SW HM [80] vo verzii 17.0 x64. V tabuľke 15 sú zobrazené parametre kodéru a v tabuľke 16 parametre dekodéru HEIC. Kompresia a dekompresia kodekom HEIC:

```
TAppEncoder -c c:\HEIC\cfg\encoder_intra_main.cfg -i <názov>.yuv -b  
<názov>.bin -o <názov>.yuv -wdt <číslo> -hgt <číslo> --InputBitDepth=8  
--OutputBitDepth=8 --InputChromaFormat=444 -cf 420 -fr 1 -f 1  
--Profile=main --Level=6.2 -q <číslo> --VideoFullRange  
TAppDecoder -b out.bin -o out.yuv
```

Tabuľka 15 Parametre kodéru HEIC

Parameter	Funkcia parametru
TAppEncoder	Spustenie kodéru HEIC
-c	Výber konfiguračného súboru
-i	Referenčný súbor YUV
-b	Komprimovaný súbor BIN
-o	Rekonštruovaný/dekomprimovaný súbor YUV
-wdt	Šírka obrazu v pixeloch
-hgt	Výška obrazu v pixeloch
--InputBitDepth	Bitová hĺbka referenčného obrázku
--OutputBitDepth	Bitová hĺbka komprimovaného obrázku
--InputChromaFormat	Chromatické podvzorkovanie referenčného obrázku
-cf	Chromatické podvzorkovanie komprimovaného obrázku
-fr	Počet snímkov za sekundu
-f	Počet zakódovaných snímkov
--Profile	Profil, main = základný profil
--Level	Level, 6.2 = maximálne rozlíšenie 8192×4320@120 FPS
-q	Kvantizačný parameter, 0 (najkvalitnejšie) – 51 (najmenej kvalitne)
--VideoFullRange	Plný rozsah hodnôt (0 – 255)

Tabuľka 16 Parametre dekodéru HEIC

Parameter	Funkcia parametru
TAppDecoder	Spustenie dekodéru HEIC
-b	Komprimovaný súbor BIN
-o	Dekomprimovaný súbor YUV

5.4.4 Kompresia kodekom AVIF

Komprimovanie a dekomprimovanie obrázkov kodekom AVIF bolo uskutočnené referenčným SW libavif [81], [82] vo verzii 0.11.1 x64. V tabuľke 17 sú zobrazené parametre kodéru a v tabuľke 18 parametre dekodéru AVIF.

Tabuľka 17 Parametre kodéru AVIF

Parameter	Funkcia parametru
avifenc	Spustenie kodéru AVIF
-j	Počet využitých jadier/vlákien CPU, all = všetky dostupné
-d	Bitová hĺbka
-y	Chromatické podvzorkovanie
-r	Rozsah obrazových hodnôt (plný, limitovaný)
--min	Minimálny kvantizačný parameter
--max	Maximálny kvantizačný parameter
-s	Rýchlosť (kvalita), 0 (najrýchlejšie) – 10 (najkvalitnejšie)
-c	Kodek
--ignore-exif	Ignorovanie EXIF metadát
--ignore-xmp	Ignorovanie XMP metadát
--ignore-icc	Ignorovanie ICC profilu
-a	Pokročilé nastavenia
enable-chroma-deltaq	Povolenie delta kvantizácie v chromatických kanáloch
tune	Doladenie pre zvolenú metriku
<názov>.png	Vstupný/referenčný súbor
<názov>.avif	Výstupný/komprimovaný súbor

Tabuľka 18 Parametre dekodéru AVIF

Parameter	Funkcia parametru
avifdec	Spustenie dekodéru AVIF
-j	Počet využitých jadier/vlákien CPU, all = všetky dostupné
-c	Kodek
-d	Bitová hĺbka výstupného PNG súboru
--png-compress	PNG kompresia, 0 = bez kompresie
-u	Farebné nadvzorkovanie, best = najvyššia kvalita
--	Koniec parametrov
<názov>.avif	Vstupný/komprimovaný súbor
<názov>.png	Výstupný/dekomprimovaný súbor

Kompresia a dekompresia kodekom AVIF:

```
avifenc -j all -d 8 -y 420 -r f --min <číslo> --max <číslo> -s <číslo>
-c aom --ignore-exif --ignore-xmp --ignore-icc -a
enable-chroma-deltaq=1 -a tune=ssim <názov>.png <názov>.avif

avifdec -j all -c aom -d 8 --png-compress 0 -u best -- <názov>.avif
<názov>.png
```

5.4.5 Kompresia kodekom JPEG XL

Komprimovanie a dekomprimovanie obrázkov kodekom JPEG XL bolo uskutočnené referenčným SW libjxl [83] vo verzii 0.7.0 x64. V tabuľke 19 sú zobrazené parametre kodéru a v tabuľke 20 parametre dekodéru JPEG XL. Kompresia a dekompresia kodekom JPEG XL:

```
cjxl <názov>.png <názov>.jxl -q <číslo> -e 9 --brotli_effort 11
--num_threads=-1 -P 15

djxl <názov>.jxl <názov>.png --num_threads=12--bits_per_sample=8
```

Tabuľka 19 Parametre kodéru JPEG XL

Parameter	Funkcia parametru
cjxl	Spustenie kodéru JPEG XL
<názov>.png	Vstupný/referenčný súbor
<názov>.jxl	Výstupný/komprimovaný súbor
-q	Kvalita, 0 (najnižšia) – 100 (najvyššia)
-e	Komplexnosť, 1 (najnižšia) – 9 (najvyššia)
--brotli_effort	Brotli komplexnosť, 0 (najnižšia) – 11 (najvyššia)
--num_threads	Počet využitých jadier/vláskien CPU, -1 = predvolená hodnota PC
-P	Modulárny prediktor, 15 = najvyššia komplexnosť

Tabuľka 20 Parametre dekodéru JPEG XL

Parameter	Funkcia parametru
djxl	Spustenie dekodéru JPEG XL
<názov>.jxl	Vstupný/komprimovaný súbor
<názov>.png	Výstupný/dekomprimovaný súbor
--num_threads	Počet využitých jadier/vláskien CPU
--bits_per_sample	Bitová hĺbka výstupného PNG súboru

5.4.6 Kompresia kodekom VVC Intra

Komprimovanie a dekomprimovanie obrázkov kodekom VVC Intra bolo otestované referenčným SW VTM [84] vo verzii 18.1 x64 a nereferenčným SW VVenC [85] vo verzii 1.7.0 x64 a VVdeC [86] vo verzii 1.6.0 x64. Rozdiel medzi VTM a VVenC je ten,

že v kodéri VVenC nie sú na rozdiel od VTM implementované všetky funkcie kodeku VVC Intra, avšak kompresia s použitím VVenC je mnohonásobne rýchlejšia, pričom rozdiel v kvalite je veľmi malý [87]. Preto bol pre kompresiu kodekom VVC Intra použitý SW VVenC a VVdeC pre dekompresiu. V tabuľke 21 sú zobrazené parametre kodéru a v tabuľke 22 parametre dekodéru VVC Intra. Kompresia a dekompresia kodekom VVC Intra (VVenC a VVdeC):

```
vvencapp -i in420.yuv -s 512x256 -c yuv420 -r 1 -f 1 -o outvvenc.266
--preset=slower -b 0 -q 5 -t 8 --tiles 4x4
--profile=main_10_still_picture --level=6.2

vvdecapp -b outvvenc.266 -o outvvenc420p10le.yuv -t -1
```

Tabuľka 21 Parametre kodéru VVC Intra

Parameter	Funkcia parametru
vvencapp	Spustenie kodéru VVC Intra
-i	Referenčný súbor YUV
-s	Rozlíšenie obrazu v pixeloch
-c	Chromatické podvzorkovanie komprimovaného obrázku
-r	Snímková frekvencia
-f	Počet zakódovaných snímkov
-o	Komprimovaný súbor 266
--preset	Nastavenie rýchlosti, slower = najpomalšie a najkvalitnejšie
-b	Dátový tok, 0 = konštantný kvantizačný parameter
-q	Kvantizačný parameter, 0 (najkvalitnejšie) – 63 (najmenej kvalitne)
-t	Počet CPU vlákien, odporúčané 8 pre rozlíšenie aspoň 720p
--tiles	Rozdelenie na dlaždice
--profile	Profil, main_10_still_picture = základný profil, 10 bitov na kanál
--level	Level, 6.2 = maximálne rozlíšenie 8192×4320@120 FPS

Tabuľka 22 Parametre dekodéru VVC Intra

Parameter	Funkcia parametru
vvdecapp	Spustenie dekodéru VVC Intra
-b	Komprimovaný súbor 266
-o	Dekomprimovaný súbor YUV
-t	Počet CPU vlákien, 0 = automatická detekcia

5.5 Softwarové nástroje umelej inteligencie

Existuje množstvo nástrojov na prácu so strojovým učením, neurónovými sieťami a hlbokým učením. Medzi najznámejšie patria TensorFlow [88] od tímu Google Brain zo spoločnosti Google, PyTorch [89] od tímu Meta AI zo spoločnosti Meta (Facebook),

alebo MATLAB Deep Learning Toolbox od spoločnosti MathWorks. Softwarom ONNX je možné konvertovať natrénované modely medzi jednotlivými nástrojmi. Najčastejšie používanými nástrojmi sú práve TensorFlow a PyTorch.

5.5.1 Inštalácia, nastavenie a sprevádzkovanie nástroja TensorFlow

TensorFlow obsahuje podporu pre programovacie jazyky Python [90] a C++. Odporúčaný je jazyk Python. Odporúčaný nástroj pre vývoj je Miniconda, v tomto prípade je použitá Anaconda, pretože obsahuje väčšie množstvo knižníc, modulov a prívetivejšie používateľské prostredie, ktoré nebude potrebné inštalovať dodatočne. Súčasťou nástroja Anaconda je aj programovacie prostredie Spyder. Pre paralelizáciu výpočtov pomocou NVIDIA GPU je potrebné nainštalovať ovládač grafickej karty, vývojársky kit CUDA, nástroj umožňujúci ML a DL aplikácie cuDNN a nástroj umožňujúci využiť Tensor jadrá TensorRT. Cez príkazový riadok nástroja Anaconda je potrebné vytvoriť pracovné prostredie, aktivovať ho, nainštalovať doň vývojársky kit CUDA, nástroj cuDNN, nainštalovať GPU verziu nástroja TensorFlow, do pracovného prostredia doinštalovať TensorRT moduly, ktoré sú súčasťou nástroja TensorRT a modul PyCUDA. V tabuľke 23 sú jednotlivé nástroje zoradené v poradí, v akom ich je potrebné inštalovať a súčasťou je aj použitá verzia a za tabuľkou sú zobrazené príkazy pre vytvorenie pracovného prostredia, inštaláciu nástroja TensorFlow a overenie funkčnosti.

Tabuľka 23 Nástroje pre TensorFlow

Nástroj	Verzia
Ovládač grafickej karty	NVIDIA Display Driver 527.34 win10 x64 [91]
Python	Python 3.10.8
Anaconda	Anaconda3 2022.10 x64 [92]
Vývojársky kit CUDA	CUDA Toolkit 11.2.0 win10 x64 [93]
Nástroj cuDNN	cuDNN 8.1.0.77 x64 [94]
Nástroj TensorRT	TensorRT 7.2.3.4 x64 [95]
Python CUDA	cuda-toolkit 11.2
Python cuDNN	cuda-cudnn 8.1.0
TensorFlow s podporou GPU	tensorflow-gpu 2.10.0 cp310 win x64 [88]
TensorRT moduly	graphsurgeon-0.4.5-py2.py3-none-any.whl [95] uff-0.6.9-py2.py3-none-any.whl [95] onnx_graphsurgeon-0.2.6-py2.py3-none-any.whl [95]
PyCUDA	pycuda 2019.1.1 [96]

```
conda create --name tf python=3.10
conda activate tf
conda install -c conda-forge cudatoolkit=11.2 cudnn=8.1.0
pip install --upgrade pip
python.exe -m pip install --upgrade pip
pip install tensorflow_gpu-2.10.0-cp310-cp310-win_amd64.whl
python -c "import tensorflow as tf;
print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
python -c "import tensorflow as tf;
print(tf.config.list_physical_devices('GPU'))"
python -m pip install c:\TensorRT-7.2.3.4\graphsurgeon\
graphsurgeon-0.4.5-py2.py3-none-any.whl
python -m pip install c:\TensorRT-7.2.3.4\uff\
uff-0.6.9-py2.py3-none-any.whl
python -m pip install c:\TensorRT-7.2.3.4\onnx_graphsurgeon\
onnx_graphsurgeon-0.2.6-py2.py3-none-any.whl
pip install pycuda==2019.1.1
```

5.5.2 Inštalácia, nastavenie a sprevádzkovanie nástroja PyTorch

PyTorch obsahuje podporu pre programovacie jazyky Python a C++. Boli použité rovnaké nástroje ako pri nástroji TensorFlow, teda Anaconda a Python. Rovnako je potrebný nainštalovaný ovládač grafickej karty, vývojársky kit CUDA, prípadne nástroje cuDNN a TensorRT. Cez príkazový riadok nástroja Anaconda je potrebné vytvoriť pracovné prostredie, aktivovať ho, nainštalovať doň Python, PyTorch a vývojársky kit CUDA. V tabuľke 24 sú jednotlivé nástroje zoradené v poradí, v akom ich je potrebné inštalovať a súčasťou je aj použitá verzia a za tabuľkou sú zobrazené príkazy pre vytvorenie pracovného prostredia, inštaláciu nástroja PyTorch a overenie funkčnosti.

Tabuľka 24 Nástroje pre PyTorch

Nástroj	Verzia
Ovládač grafickej karty	NVIDIA Display Driver 527.34 win10 x64 [88]
Anaconda	Anaconda3 2022.10 x64 [89]
Python	python 3.10.8
PyTorch	pytorch 1.13.0 [86]
Python CUDA	cudatoolkit 11.7

```
conda create --name PyTorch1130 python=3.10
conda activate PyTorch1130
conda install pytorch torchvision torchaudio pytorch-cuda=11.7
-c pytorch -c nvidia
cd c:\ProgramData\Anaconda3\envs\PyTorch1130\
python
import torch
x = torch.rand(5, 3)
print(x)
torch.cuda.is_available()
```

5.6 Kompresia obrazu s využitím algoritmov strojového a hlbokého učenia

Pre kompresiu obrazu pomocou ML a DL algoritmov sú využité štyri algoritmy, z ktorých tri sú zamerané na kompresiu planárneho obrazu a jeden je zameraný na kompresiu 360° obrazu. Avšak aj týmito algoritmami zameranými na planárne obrázky je možné komprimovať 360° obrázky v ERP projekcii. Využité modely sú natréňované modely jednotlivých algoritmov. Všetky použité implementácie DL algoritmov sú rovnako v najaktuálnejších verziách.

5.6.1 Algoritmus Full Resolution Image Compression with Recurrent Neural Networks

Použitý model je natréňovaný model residual_gru.pb [97]. Pre správnu funkciu algoritmu je potrebné prostredie obsahujúce moduly zobrazené v tabuľke 25, za ktorou sú uvedené príkazy pre kompresiu a dekompresiu a v tabuľke 26 sú zobrazené parametre algoritmu.

Tabuľka 25 Nástroje algoritmu FRICwRNN

Nástroj	Verzia
Python	python 3.6
Python CUDA	cuda toolkit 10
Python cuDNN	cudnn 7.6.0
TensorFlow s podporou GPU	tensorflow-gpu 1.15.0 cp36 win x64
Knižnica protobuf	protobuf 3.6.0
Knižnica numpy	numpy 1.19.3

```
python encoder.py --input_image=<názov>.png --output_codes=<názov>.npz  
--iteration=<číslo> --model=residual_gru.pb  
python decoder.py --input_codes=<názov>.npz --output_directory=<názov>  
--model=residual_gru.pb
```

Tabuľka 26 Parametre algoritmu FRICwRNN

Parameter	Funkcia parametru
python	Spustenie nástroja python
encoder.py	Spustenie kodéra
--input_image	Vstupný/referenčný súbor
--output_codes	Výstupný/komprimovaný kód
--iteration	Kvalita (BPP), 0 (najnižšia) – 15 (najvyššia)
--model	Natréňovaný model
decoder.py	Spustenie dekodéra
--input_codes	Vstupný/komprimovaný kód
--output_directory	Adresár výstupných dekomprimovaných súborov

5.6.2 Algoritmus Context-adaptive Entropy Model for End-to-end Optimized Image Compression

Použitý model je natrénovaný model [98] optimalizovaný pre metriku MS-SSIM. Pre správnu funkciu algoritmu je potrebné prostredie obsahujúce moduly zobrazené v tabuľke 27, za ktorou sú uvedené príkazy pre kompresiu a dekompresiu a v tabuľke 28 sú zobrazené parametre algoritmu.

Tabuľka 27 Nástroje algoritmu CAEMfe2EOIC

Nástroj	Verzia
Python	python 3.6
Python CUDA	cuda toolkit 10
Python cuDNN	cuda 7.6.0
TensorFlow s podporou GPU	tensorflow-gpu 1.15.0 cp36 win x64
Knižnica scipy	scipy 1.5.4
Knižnica requests	requests 2.27.1
Knižnica pillow	pillow 8.4.0
Knižnica Hdf5	hdf5 1.10.1

```
python encode.py --model_type <číslo> --input_path ./<názov>.png  
--quality_level <číslo>  
python decode.py --compressed_file_path ./<názov>.cmp
```

Tabuľka 28 Parametre algoritmu CAEMfe2EOIC

Parameter	Funkcia parametru
python	Spustenie nástroja python
encode.py	Spustenie kódéra
--model_type	Typ optimalizovaného natrénovaného modelu
--input_path	Vstupný/referenčný súbor
--quality_level	Kvalita (BPP), 1 (najnižšia) – 9 (najvyššia)
decode.py	Spustenie dekodéra
--compressed_file_path	Umiestnenie komprimovaného kódu

5.6.3 Algoritmus High-Fidelity Generative Image Compression

Použité modely sú všetky tri dostupné modely [99] určené pre tri stupne kompresie. Pre správnu funkciu algoritmu je potrebné prostredie obsahujúce moduly zobrazené v tabuľke 29, za ktorou sú uvedené príkazy pre kompresiu a dekompresiu a v tabuľke 30 sú zobrazené parametre algoritmu.

Tabuľka 29 Nástroje algoritmu HiFiC

Nástroj	Verzia
Python	python 3.7
Python CUDA	cuda toolkit 10.0
Python cuDNN	cudnn 7.6.5
PyTorch s podporou GPU	torch 1.6.0+cu101 cp37 win x64
Torchvision	torchvision 0.7.0+cu101 cp37 win x64
Pandas	pandas 1.3.5
Tables	tables 3.7.0
requirements.txt	balík ďalších podporných knižníc

```
python compress.py -i data\originals\ -ckpt models\hific_low.pt
--reconstruct
python compress.py -i data\originals\ -ckpt models\hific_med.pt
--reconstruct
python compress.py -i data\originals\ -ckpt models\hific_hi.pt
--reconstruct
python compress.py -i data\originals\ -ckpt models\hific_low.pt --save
python compress.py -i data\originals\ -ckpt models\hific_med.pt --save
python compress.py -i data\originals\ -ckpt models\hific_hi.pt --save
```

Tabuľka 30 Parametre algoritmu HiFiC

Parameter	Funkcia parametru
python	Spustenie nástroja python
compress.py	Spustenie kódéra
-i	Adresár vstupných/referenčných súborov
-ckpt	Typ natrénovaného modelu
--reconstruct	Rekonštrukcia bez entropického kódéra
--save	Použitie rANS entropického kódéra a uloženie komprimovaných kódov

5.6.4 Algoritmus End-to-End Optimized 360° Image Compression

Keďže algoritmus je určený na kompresiu výrezov z obrázka s rozlíšením 1K (1024×512) pixelov, tieto výrezy boli vytvorené pomocou programu FastStone Image Viewer [75]. Použitý model je natrénovaný model [100] optimalizovaný pre metriku VSSIM. Pre správnu funkciu algoritmu je potrebné prostredie obsahujúce moduly zobrazené v tabuľke 31, za ktorou sú uvedené príkazy pre kompresiu a dekompresiu a v tabuľke 32 sú zobrazené parametre algoritmu.

Tabuľka 31 Nástroje algoritmu LIC360

Nástroj	Verzia
Python	python 3.7
PyTorch s podporou GPU	pytorch 1.13.1 cuda 11.7
Knižnica cv2 (python-opencv)	opencv-python 4.6.0.66
Knižnica numpy	numpy 1.19.3

```
python test\lic360_demo.py --enc --img-list test\<>názov>.png
--code-list test\<>názov> --model-idx <číslo> --ssim
python test\lic360_demo.py --dec --out-list test\<>názov>.png
--code-list test\<>názov> --model-idx <číslo> --ssim
```

Tabuľka 32 Parametre algoritmu LIC360

Parameter	Funkcia parametru
python	Spustenie nástroja python
test\lic360_demo.py	Spustenie algoritmu
--enc	Spustenie kodéra
--img-file	Vstupné/referenčné súbory
--code-list	Výstupné/komprimované kódy
--model-idx	Model s natrénovanou kvalitou BPP, 1 (najnižšia) – 8 (najvyššia)
--ssim	Vynútenie VSSIM optimalizovaného modelu, predvolene VMSE
--dec	Spustenie dekodéra
--out-list	Výstupné/dekomprimované súbory

5.7 Výsledky kompresie a dekompresie

Vo výsledkoch kompresie a dekompresie sú uvedené maximálne rozlíšenia obrázkov, ktoré bolo možné komprimovať a dekomprimovať jednotlivými kodekmi. Sú tu zahrnuté HW nároky jednotlivých kodekov na CPU, GPU, RAM a VRAM (pamäť GPU) a časy potrebné na kompresiu a dekompresiu obrázkov. Celkový počet vytvorených komprimovaných obrázkov je 4110. Tieto výsledky sú publikované aj v článku [101]. Ukážku kompresných artefaktov spôsobených jednotlivými kodekmi na časti jedného z obrázkov je možné vidieť v prílohe B, kde sú označené aj nastavenia kvality kompresie každého kodeku.

5.7.1 Hardwarové nároky

Na už zmienenom notebooku bolo možné skomprimovať obrázky v 8K (8192×4096) rozlíšení len konvenčnými kodekmi. Kvôli vysokým nárokom DL algoritmov na RAM a VRAM bolo možné skomprimovať obrázky len v nižších rozlíšeníach. Pri algoritmoch FRICwRNN a HiFiC bolo maximálne dosiahnuteľné rozlíšenie 2K (2048×1024) a pri algoritme CAEMfE2EOIC bolo maximálne dosiahnuteľné rozlíšenie 4K (4096×2048). Pri vyšších rozlíšeníach nastal problém s nedostatkom pamäte RAM, prípadne VRAM

a tento problém nebolo možné vyriešiť ani dostatočne veľkým stránkovacím súborom na SSD. Ako už bolo spomenuté, algoritmom LIC360 sú obrázky komprimované len v 1K rozlíšení a v tomto prípade žiaden problém pri kompresii nenastal.

5.7.2 Implementácie využívajúce CPU a GPU

Implementácie konvenčných kodekov využívajú pre kompresiu a dekompresiu len CPU. DL algoritmy FRICwRNN, HiFiC a LIC360 sú optimalizované pre využitie GPU pri kompresii aj dekompresii, vplyvom čoho je celý proces značne urýchlený. Pokiaľ by nebolo možné využiť GPU, čas kompresie a dekompresie by bol dlhší. To je aj prípad DL algoritmu CAEMfE2EOIC, ktorý nie je optimalizovaný na využitie GPU a obrázky sú komprimované a dekomprimované len pomocou CPU.

5.7.3 Využitie hardwarových prostriedkov

Pri kompresii obrázkov všetkými konvenčnými kodekmi a algoritmom CAEMfE2EOIC boli využité len CPU a RAM. Pri kompresii algoritmami FRICwRNN, HiFiC a LIC360 boli okrem CPU a RAM využité aj GPU a VRAM. Prehľadné zobrazenie využitia HW prostriedkov pri kompresii je v tabuľke 33.

Tabuľka 33 Využitie hardwarových prostriedkov pri kompresii

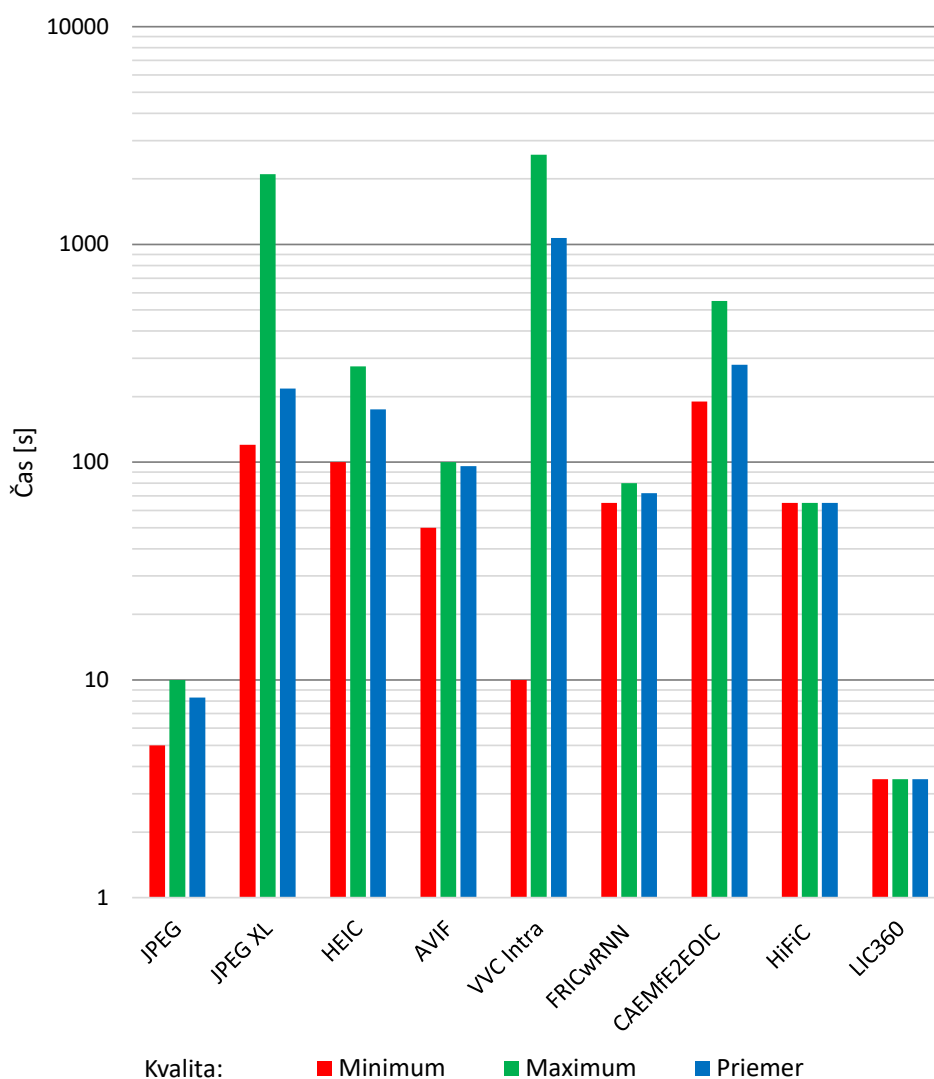
Kodek (Rozlíšenie)	CPU [%]	GPU [%]	RAM [GB]	VRAM [GB]
JPEG (8K)	17	0	0,1	0
HEIC (8K)	16	0	0,1	0
AVIF (8K)	100	0	0,7	0
JPEG XL (8K)	25	0	8,8	0
VVC Intra (8K)	100	0	0,8	0
FRICwRNN (2K)	15	100	2,5	7,2
CAEMfE2EOIC (4K)	16	0	6,1	0
HiFiC (2K)	25	100	2,8	7,3
LIC360 (1K)	25	100	2,4	3,5

Z tabuľky je vidno, že pre kompresiu konvenčnými kodekmi JPEG, HEIC, AVIF a VVC Intra je využité iba minimum RAM, ktoré je menej ako 1 GB. Výnimkou je len kodek JPEG XL, pre ktorý je potrebných 8,8 GB RAM. Kodeky AVIF a VVC Intra sú navyše veľmi efektívne paralelizované a pri ich činnosti sú naplno využité všetky dostupné jadrá CPU, čo urýchľuje kompresiu. Ostatné konvenčné kodeky dokázali využiť len jedno, prípadne dve CPU jadrá, čoho výsledkom je dlhší čas kompresie. Rovnako aj DL algoritmus CAEMfE2EOIC dokázal pri kompresii využiť iba jedno CPU jadro. Využitie RAM bolo 6,1 GB, čo je pomerne vysoké číslo vzhľadom na 4K rozlíšenie. Pre činnosť ostatných DL algoritmov FRICwRNN, HiFiC a LIC360 bolo v pomere k rozlíšeniu potrebnej ešte viac pamäte, v tomto prípade aj VRAM, pretože kompresia

prebiehala hlavne pomocou GPU, ktorej využitie bolo 100%. Tento nárast je však pochopiteľný z dôvodu vysokej paralelizácie GPU operácií. Pri pohľade na využitie RAM sa môže zdať, že kompresia by mala prebehnúť aj vo vyššom rozlíšení bez väčších problémov, avšak s nárastom rozlíšenia stúpalo množstvo potrebnej pamäte omnoho rýchlejšie ako samotné rozlíšenie.

5.7.4 Časy kompresie a dekompresie

V tabuľke 34 sú uvedené časy kompresie pre jednotlivé kodeky pri kompresii s najnižšou obrazovou kvalitou, najvyššou obrazovou kvalitou a priemerom všetkých dostupných obrazových kvalít, respektíve kvantizačných parametrov každého kodeku. Rovnako sú tieto časy zobrazené aj na obrázku 29. Tieto výsledky sú priemerom cez všetky obrázky z datasetu.



Obrázok 29 Časy kompresie

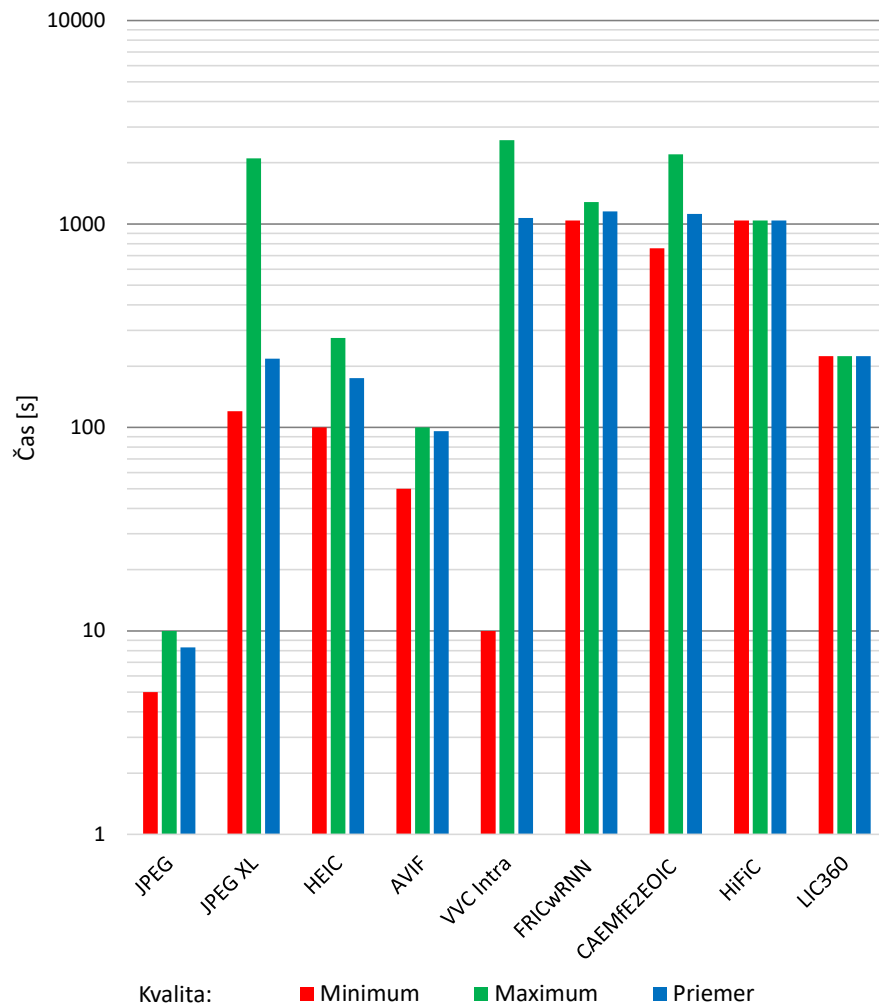
Tabuľka 34 Časy kompresie

Kodek (Rozlíšenie)	Čas kompresie [s]		
	Najnižšia kvalita	Najvyššia kvalita	Priemer
JPEG (8K)	5	10	8,3
HEIC (8K)	100	275	175
AVIF (8K)	50	100	96
JPEG XL (8K)	120	2100	218
VVC Intra (8K)	10	2580	1069
FRICwRNN (2K)	65	80	72
CAEMfE2EOIC (4K)	190	550	280
HiFiC (2K)	65	65	65
LIC360 (1K)	3,5	3,5	3,5

Všetky konvenčné kodeky boli nastavené na kompresiu s najvyššou komplexnosťou, aby bola zaručená čo najvyššia efektivita kompresie. To má výrazný dopad na čas kompresie. Z tabuľky je vidno, že najrýchlejšie boli obrázky komprimované DL algoritmom LIC360. Je to z dôvodu nízkeho rozlíšenia a využitia GPU. Ďalší v poradí je kodek JPEG, ktorý je už na dnešné pomery výpočetne nenáročný. V prípade priemerných výsledkov nasleduje HiFiC a FRICwRNN. Kompresia kodekom HEIC by mala byť rýchlejšia ako kodekom AVIF, avšak kvôli už zmienenej optimalizácii využitia všetkých dostupných CPU jadier kodekom AVIF bola kompresia týmto kodekom rýchlejšia, ako kodekom HEIC. Ďalším v poradí je kodek JPEG XL. Je dôležité dodať, že tento kodek ešte nie je úplne dokončený a optimalizovaný a nie sú implementované všetky jeho funkcionality. Po kodeku JPEG XL nasleduje DL algoritmus CAEMfE2EOIC a najdlhší priemerný čas kompresie bol dosiahnutý kodekom VVC Intra.

Ďalší spozorovaný fakt je, že pri kompresii DL algoritmi HiFiC a LIC360 nenastal rozdiel v čase kompresie pri zmene kvality. Môže to byť spôsobené optimalizáciou pre GPU. V prípade DL algoritmu FRICwRNN bola zmena kompresného času len minimálna. Čas kompresie v prípade DL algoritmu CAEMfE2EOIC a všetkých konvenčných kodekov bol značne závislý na nastavenej kvalite kompresie. Ten bol podľa predpokladu najnižší pre kompresiu s najnižšou kvalitou a najvyšší pre kompresiu s najvyššou kvalitou.

Keďže sú v tabuľke porovnávané časy kompresie rozličnými kodekmi pre rôzne rozlíšenia, je vhodné zjednotiť časy kompresie pre jedno rozlíšenie spoločne pre všetky kodeky. Kvôli potrebe vysokého rozlíšenia 360° obrazu sú hodnoty času kompresie pre DL algoritmy odhadnuté pre rozlíšenie 8K a zobrazené v tabuľke 35 spolu s reálnymi časmi kompresie konvenčnými kodekmi. Grafické znázornenie sa nachádza na obrázku 30. 8K rozlíšenie by pravdepodobne bolo možné dosiahnuť s využitím počítača s dostatočnou kapacitou RAM a VRAM.



Obrázok 30 Časy kompresie pre 8K rozlíšenie

Tabuľka 35 Časy kompresie pre 8K rozlíšenie

Kodek	Čas kompresie [s]		
	Najnižšia kvalita	Najvyššia kvalita	Priemer
JPEG	5	10	8,3
HEIC	100	275	175
AVIF	50	100	96
JPEG XL	120	2100	218
VVC Intra	10	2580	1069
FRICwRNN	1040	1280	1152
CAEMfE2EOIC	760	2200	1120
HiFiC	1040	1040	1040
LIC360	224	224	224

Po odhade časov kompresie pre 8K rozlíšenie v prípade DL algoritmov by bolo poradie čiastočne zmenené. V prípade priemerného času kompresie by bol najkratší čas dosiahnutý kodekom JPEG. Ďalšie v poradí by boli kodeky AVIF, HEIC a JPEG XL. Nasledovali by algoritmy LIC360 a HiFiC, kodek VVC Intra a algoritmus CAEMfE2EOIC. Najdlhší čas kompresie by bol dosiahnutý algoritmom FRICwRNN. V prípade použitia referenčného SW VTM pre kompresiu kodekom VVC Intra miesto VVenC by bol čas kompresie najdlhší práve týmto kodekom.

Čas dekompresie obrázkov komprimovaných konvenčnými kodekmi bol v rozsahu do pár sekúnd, z čoho najrýchlejší kodek bol podľa predpokladu znova JPEG. Čo sa týka dekompresie obrázkov komprimovaných DL algoritmami, časy dekompresie boli zhodné s časmi kompresie. To je dané architektúrou použitých neurónových sietí, kde architektúra dekodéru je reverzná k architektúre kodéru, čo je aj princíp autoenkodérov.

6. HODNOTENIE KVALITY OBRAZU

Pre hodnotenie kvality komprimovaných obrázkov je použitých 7 plne referenčných objektívnych metrík a 1 subjektívna metrika. Výsledky všetkých metrík spolu s rozsiahlymi tabuľkami a grafmi sú súčasťou elektronickej prílohy.

6.1 Implementácia objektívnych metrík

Na objektívne vyhodnotenie kvality obrázkov boli použité metriky PSNR-HVS-M, WS-PSNR, MS-SSIM, VIFp, FSIMc, GMSD a VMAF. Výsledky objektívneho hodnotenia kvality dosiahnuté metrikami PSNR-HVS-M, MS-SSIM, VIFp, FSIMc, GMSD a VMAF sú publikované v článku [102].

6.1.1 Metriky PSNR-HVS-M, MS-SSIM a VIFp

Na výpočet kvality týmito metrikami bol použitý program VQMT [103] vo verzii 1.1 x64 s potrebným balíkom knižníc OpenCV, ktoré sú dodávané spolu s programom. Vstupným formátom je formát YUV. V tabuľke 36 sa nachádzajú parametre príkazu použitého na výpočet kvality a za tabuľkou sa nachádza samotný príkaz.

Tabuľka 36 Parametre programu VQMT

Parameter	Funkcia
VQMT	Spustenie programu VQMT
<referenčný>	Referenčný súbor YUV
<skreslený>	Skreslený súbor YUV
W	Šírka obrazu v pixeloch
H	Výška obrazu v pixeloch
1	Počet snímkov
3	Chromatické podvzorkovanie YUV444
<názov>	Názov súborov výsledných skóre jednotlivých metrík
PSNRHVSM	Použitie metriky PSNR-HVS-M
MSSSIM	Použitie metriky MS-SSIM
VIFP	Použitie metriky VIFp

```
VQMT <referenčný>.yuv <skreslený>.yuv W H 1 3 <názov> PSNRHVSM MSSSIM VIFP
```

6.1.2 Metrika WS-PSNR

Na výpočet kvality touto metrikou bol použitý program 360tools_metric [104]. Pri použití metriky WS-PSNR je projekcia obrazu ERP a obrázky z datasetu sú rovnako v tomto formáte. Vstupným formátom obrázkov je formát YUV420. V tabuľke 37 sa nachádzajú

parametre príkazu použitého na výpočet kvality a za tabuľkou sa nachádza samotný príkaz.

Tabuľka 37 Parametre programu 360tools_metric

Parameter	Funkcia
360tools_metric	Spustenie programu 360tools_metric
-o	Referenčný súbor YUV
-r	Skreslený súbor YUV
-w	Šírka obrazu v pixeloch
-h	Výška obrazu v pixeloch
-n	Počet snímkov
-q	Metrika, 3 = WS-PSNR
-f	Formát projekcie referenčného obrázku, 1 = ERP
-t	Formát projekcie skresleného obrázku, 1 = ERP
-x	Chromatické podvzorkovanie referenčného obrázku, 1 = 4:2:0
-y	Chromatické podvzorkovanie skresleného obrázku, 1 = 4:2:0
> <názov>	Názov súboru výsledného skóre metriky

```
360tools_metric -o <názov>.yuv -r <názov>.yuv -w <číslo> -h <číslo>
-n 1 -q 3 -f 1 -t 1 -x 1 -y 1 > <názov>
```

6.1.3 Metriky FSIMc a GMSD

Na výpočet kvality metrikami FSIMc a GMSD bol použitý program MATLAB [105] vo verzii R2022b x64 a MATLAB skripty vytvorené tvorcami metrik FSIMc [106] a GMSD [107]. Vstupnými parametrami sú referenčný a skreslený obrázok. Bol vytvorený pomocný MATLAB skript, ktorým sú načítavané jednotlivé obrázky a volaná funkcia výpočtu skóre metrik s následným uložením hodnoty skóre do textového súboru. Keďže skóre metriky GMSD je počítané len na jasovej zložke, vstupným formátom sú obrázky v odtieňoch šedej vo formáte PGM. Príklad skriptu:

```
fileID = fopen('FSIMc_vysledky.txt', 'w');
imageRef = imread('in.png'); % referencny obrazok
imageDis = imread('out.png'); % skresleny obrazok
[FSIM, FSIMc] = FeatureSIM(imageRef, imageDis)
fprintf(fileID, '%1.4f\n', FSIMc);
fclose(fileID);

fileID = fopen('GMSD_vysledky.txt', 'w');
Y1 = imread('in.pgm'); % referencny obrazok
Y2 = imread('out.pgm'); % skresleny obrazok
[score, quality_map] = GMSD(Y1, Y2);
fprintf(fileID, '%1.4f\n', score);
fclose(fileID);
```

6.1.4 Metrika VMAF

Na výpočet kvality metrikou VMAF bol použitý program VMAF [108] vo verzii 2.3.1. Vstupným formátom je formát Y4M. V tabuľke 38 sa nachádzajú parametre príkazu použitého na výpočet kvality a za tabuľkou sa nachádza samotný príkaz.

Tabuľka 38 Parametre programu VMAF

Parameter	Funkcia
vmaf	Spustenie programu VMAF
-r	Referenčný súbor Y4M
-d	Skreslený súbor Y4M
-m	Model
path	Cesta k modelu, bola použitá relatívna cesta
-o	Názov súboru výsledného skóre metriky
--sub	Formát zápisu skóre, sub = titulok
--threads	Počet pracovných CPU vlákien

```
vmaf -r <názov>.y4m -d <názov>.y4m -m path="vmaf_v0.6.1.json"  
-o <názov>.txt --sub --threads <číslo>
```

6.2 Aplikácia subjektívnej metriky ACR-HR

Na overenie relevantnosti objektívneho hodnotenia a porovnania kvality obrazu medzi konvenčnými kodekmi a DL algoritmi je použité subjektívne hodnotenie. Na základe JPEG AI [46], [47] podľa odporúčania Rec. ITU-R BT.500-13, respektíve novšieho Rec. ITU-R BT.500-14 [68] a odporúčania ITU-T P.910 [69] je na subjektívne hodnotenie použitá metrika DSCQS. Avšak vzhľadom na použitie VR headsetu pri subjektívnom hodnotení kvality bola použitá metrika ACR-HR. Celkový počet respondentov bol 3, z čoho všetci respondenti sú odborníkmi v oblasti obrazovej techniky. Je to menej ako je minimum potrebné na dôkladné porovnanie výsledkov objektívnych metrik so subjektívnym vnímaním kvality, avšak na jednoduché overenie to stačí. Vzhľadom na nízke množstvo respondentov nie je meraná korelácia medzi objektívnym a subjektívnym hodnotením, ani konfidenčný interval (CI). Na výpočet tejto korelácie je v praxi používaný Pearsonov lineárny korelačný koeficient (PLCC) a Spearmanov koeficient poradovej korelácie (SROCC) [68].

Použitý VR headset bol Oculus Rift S, ktorý bol pripojený k notebooku prostredníctvom káblu DisplayPort. Pre správnu funkčnosť tohto headsetu je potrebná lokálna inštalácia SW Oculus [109] a na prezeranie obrázkov vo VR prostredí bola použitá aplikácia Oculus 360 Photos dostupná priamo v SW Oculus. Subjektívne hodnotenie prebiehalo na obrázkoch komprimovaných konvenčnými kodekmi JPEG, HEIC, AVIF, JPEG XL, VVC Intra a DL algoritmi FRICwRNN, CAEMfE2EOIC a HiFiC. Obrázky komprimované algoritmom LIC360 boli vynechané z dôvodu, že sa

jednalo len o výrezy z pôvodných obrázkov, teda hodnotenie týchto obrázkov vzhľadom na hodnotenie s použitím VR headsetu by nemalo zmysel.

Celkový počet hodnotených obrázkov bol 635, čo zodpovedá všetkým deviatim nastaveniam BPP, respektíve menej nastaveniam BPP, pokiaľ všetky tieto cieľové hodnoty BPP nebolo možné dosiahnuť. To nebolo možné z dôvodu, že DL algoritmom HiFiC je možné komprimovať obrázky iba v troch kvalitách, Algoritmom FRICwRNN je možné komprimovať obrázky v kvalitách začínajúcich od 0,1 BPP, čím je získaných celkovo sedem cieľových BPP a algoritmom LIC360 je možné komprimovať obrázky len vo ôsmich dostupných kvalitách. Zároveň, pri piatich obrázkoch nebolo možné dosiahnuť najnižšiu cieľovú hodnotu BPP kodekom JPEG XL. V týchto 635 obrázkoch sú zahrnuté aj všetky referenčné obrázky. Všetky obrázky boli náhodne očíslované a prehrávané v tomto náhodnom poradí. Vzhľadom na množstvo obrázkov trvalo subjektívne hodnotenie jedným respondentom v priemere 2 hodiny, preto bolo každé hodnotenie rozdelené na 4 sedenia po 30 minút, aby bolo zamedzené únave očí a strate sústredenia respondenta. Prezerané obrázky boli dekomprimované obrázky v PNG formáte.

6.3 Výsledky objektívnych metrík

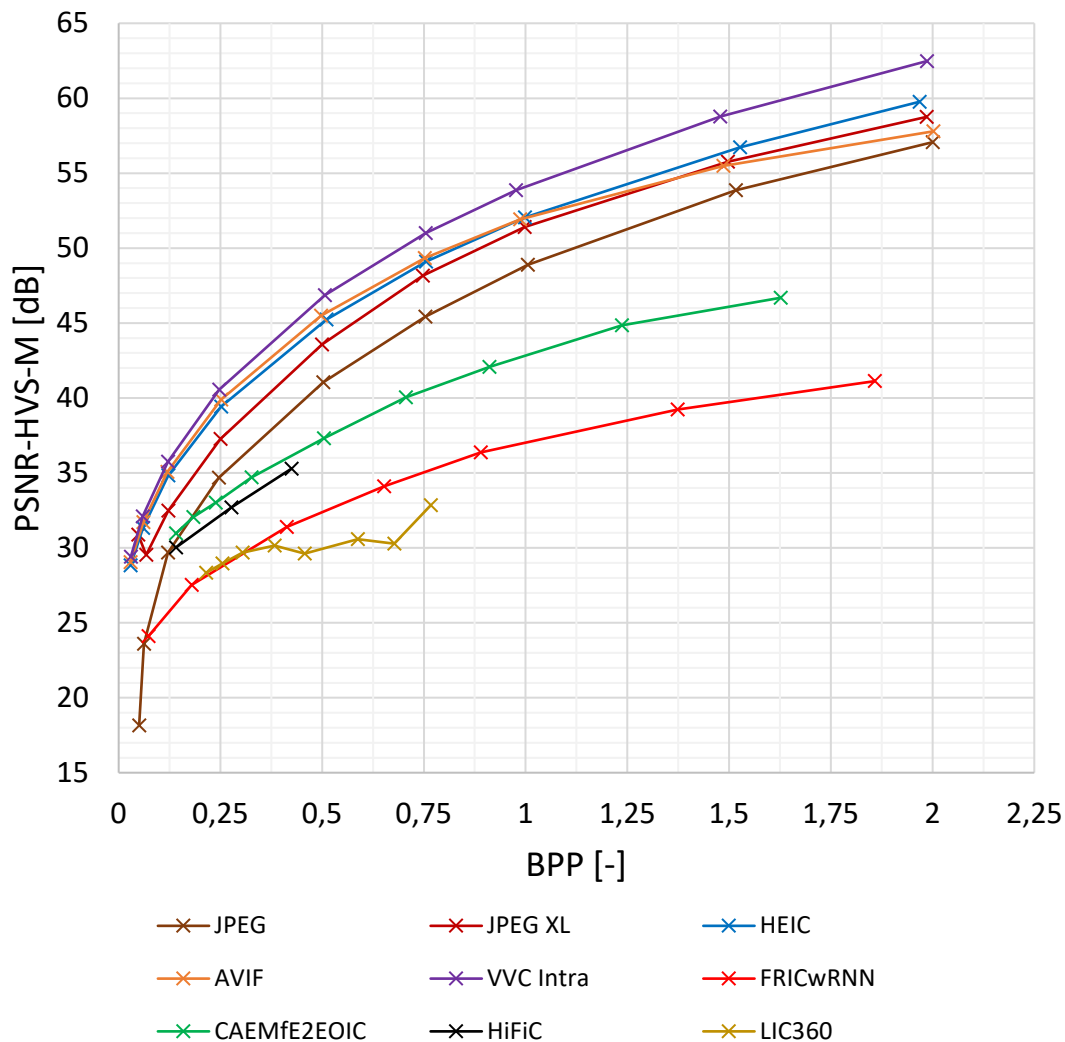
Výsledky boli zmerané pre všetkých 10 obrázkov z datasetu, ktoré boli jednotlivými kodekmi komprimované a dekomprimované v rozlíšeniach 1K, 2K, 4K a 8K. Rôzne použité rozlíšenia nie sú problémom, pretože je hodnotená schopnosť kodekov komprimovať obrázky v určitých hodnotách BPP, čo pri voľbe kvality, respektíve kvantizačného parametru, nie je hodnota závislá na rozlíšení. Zhotovené výsledky sú vo forme RD kriviek, ktoré predstavujú priemer cez všetky obrázky z datasetu pre každú metriku.

Pre porovnanie RD kriviek a vyhodnotenie kodekov z hľadiska objektívnej kvality je použitá metrika Bjøntegaard-Delta [110]. Jedná sa o porovnanie medzi RD krivkami zvoleného referenčného a testovaných kodekov. Pri tomto porovnaní sú získané hodnoty BD-PSNR a BD-Rate, ktorými je vyjadrená úspora dát v dB pre metriku BD-PSNR a v % pre metriku BD-Rate. Použitá implementácia je naprogramovaná vo forme Python skriptov [111] a z dôvodu vyššej presnosti výpočtu [111] bola vybratá metóda interpolácie Akima. Hodnoty s najvyššou a najnižšou výslednou obrazovou pre porovnanie kodekov a jednotlivé metriky sú v BD-Rate tabuľkách zvýraznené sýtou farbou.

Hodnoty indexu MS-SSIM 0,57 a 0,72 v prípade kodeku JPEG nie sú v grafe zobrazené z dôvodu lepšej čitateľnosti ostatných hodnôt. Rovnaký spôsob zachovania čitateľnosti grafu je použitý aj v prípade hodnôt indexu FSIMc, ktoré sú 0,76 a 0,88, tiež pre kodek JPEG.

6.3.1 Výsledky metriky PSNR-HVS-M

Na obrázku 31 a v tabuľke 39 sú zobrazené výsledky metriky PSNR-HVS-M.



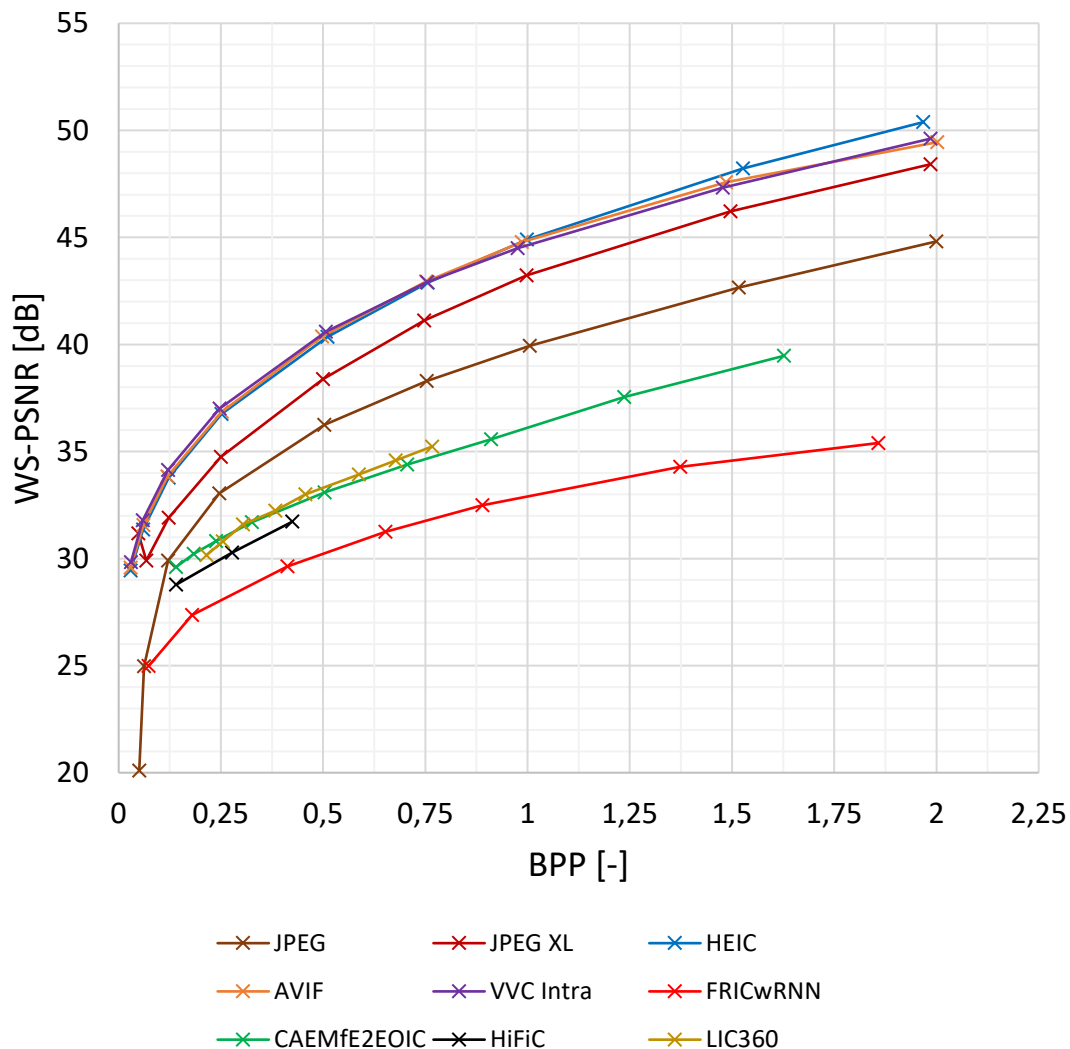
Obrázok 31 Výsledky metriky PSNR-HVS-M

Tabuľka 39 Porovnanie kompresnej účinnosti metódou BD-Rate, PSNR-HVS-M

BD-Rate [%]	JPEG	JPEG XL	HEIC	AVIF	VVC Intra	FRICwRNN	CAEMfE2EOIC	HiFiC	LIC360
JPEG	0	33	67	71	96	-60	-32	-30	-74
JPEG XL	-25	0	24	28	48	-76	-50	-53	-85
HEIC	-40	-19	0	3	20	-84	-62	-70	-91
AVIF	-42	-22	-3	0	16	-85	-64	-72	-92
VVC Intra	-49	-32	-17	-14	0	-86	-68	-75	-93
FRICwRNN	148	319	519	557	627	0	122	93	-35
CAEMfE2EOIC	47	99	164	181	214	-55	0	-20	-82
HiFiC	43	113	237	260	304	-48	25	0	-75
LIC360	290	587	1066	1175	1383	54	465	300	0

6.3.2 Výsledky metriky WS-PSNR

Na obrázku 32 a v tabuľke 40 sú zobrazené výsledky metriky WS-PSNR.



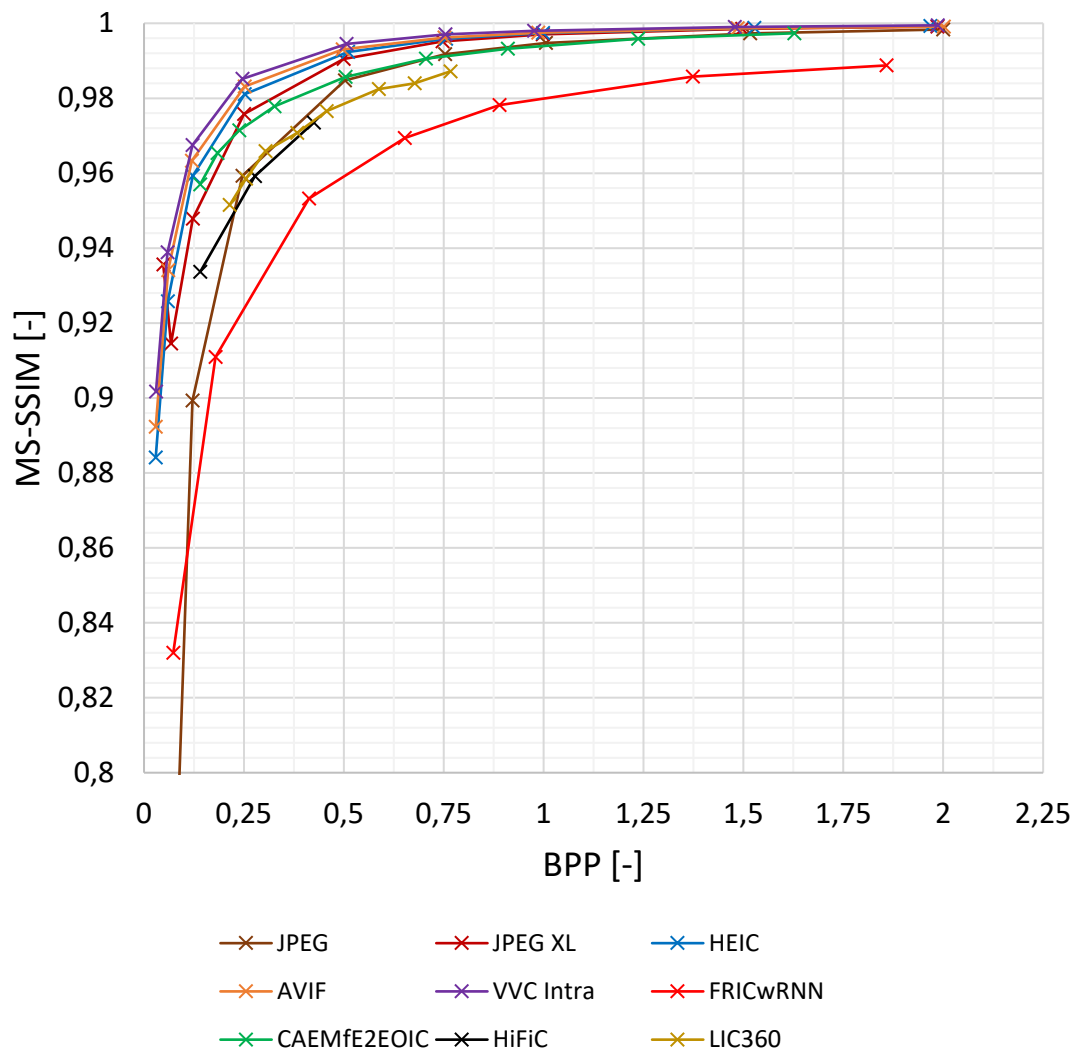
Obrázok 32 Výsledky metriky WS-PSNR

Tabuľka 40 Porovnanie kompresnej účinnosti metódou BD-Rate, WS-PSNR

BD-Rate [%]	JPEG	JPEG XL	HEIC	AVIF	VVC Intra	FRICwRNN	CAEMfE2EOIC	HiFiC	LIC360
JPEG	0	58	133	138	148	-67	-46	-50	-46
JPEG XL	-37	0	44	46	50	-84	-65	-72	-64
HEIC	-57	-31	0	1	4	-91	-78	-85	-79
AVIF	-58	-32	-1	0	3	-91	-79	-86	-80
VVC Intra	-60	-33	-4	-3	0	-92	-80	-87	-81
FRICwRNN	207	537	986	1027	1115	0	122	86	123
CAEMfE2EOIC	86	186	357	373	408	-55	0	-29	3
HiFiC	100	263	568	604	674	-46	41	0	31
LIC360	84	181	376	396	440	-55	-3	-24	0

6.3.3 Výsledky metriky MS-SSIM

Na obrázku 33 a v tabuľke 41 sú zobrazené výsledky metriky MS-SSIM.



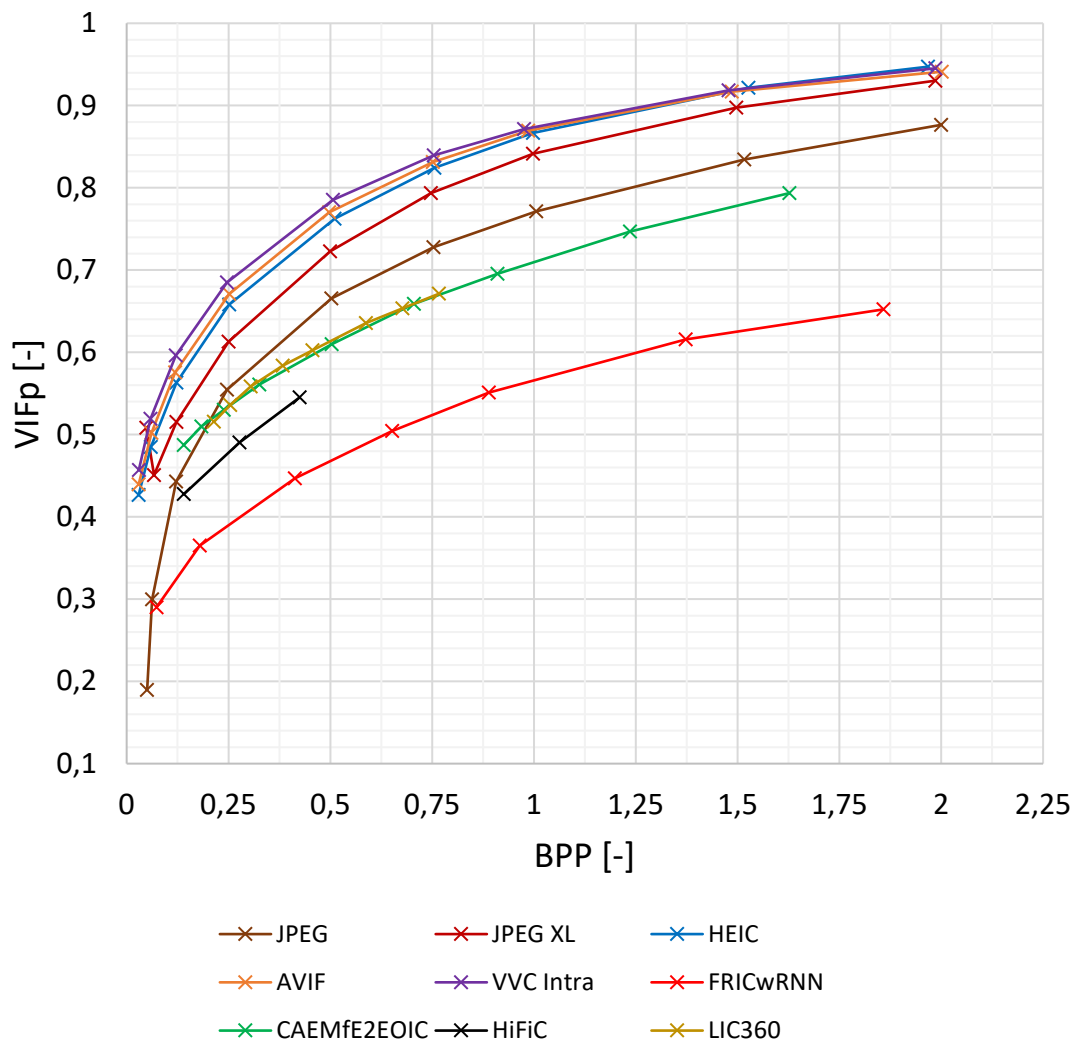
Obrázok 33 Výsledky metriky MS-SSIM

Tabuľka 41 Porovnanie kompresnej účinnosti metódou BD-Rate, MS-SSIM

BD-Rate [%]	JPEG	JPEG XL	HEIC	AVIF	VVC Intra	FRICwRNN	CAEMfE2EOIC	HiFiC	LIC360
JPEG	0	66	133	154	177	-27	23	-4	-13
JPEG XL	-40	0	28	45	65	-67	-15	-41	-42
HEIC	-57	-22	0	13	28	-75	-30	-55	-54
AVIF	-61	-31	-11	0	14	-78	-37	-60	-59
VVC Intra	-64	-39	-22	-12	0	-80	-45	-65	-63
FRICwRNN	36	202	299	351	410	0	191	79	85
CAEMfE2EOIC	-19	18	44	60	81	-66	0	-44	-36
HiFiC	4	68	120	151	183	-44	78	0	6
LIC360	15	72	117	143	174	-46	57	-6	0

6.3.4 Výsledky metriky VIFp

Na obrázku 34 a v tabuľke 42 sú zobrazené výsledky metriky VIFp.



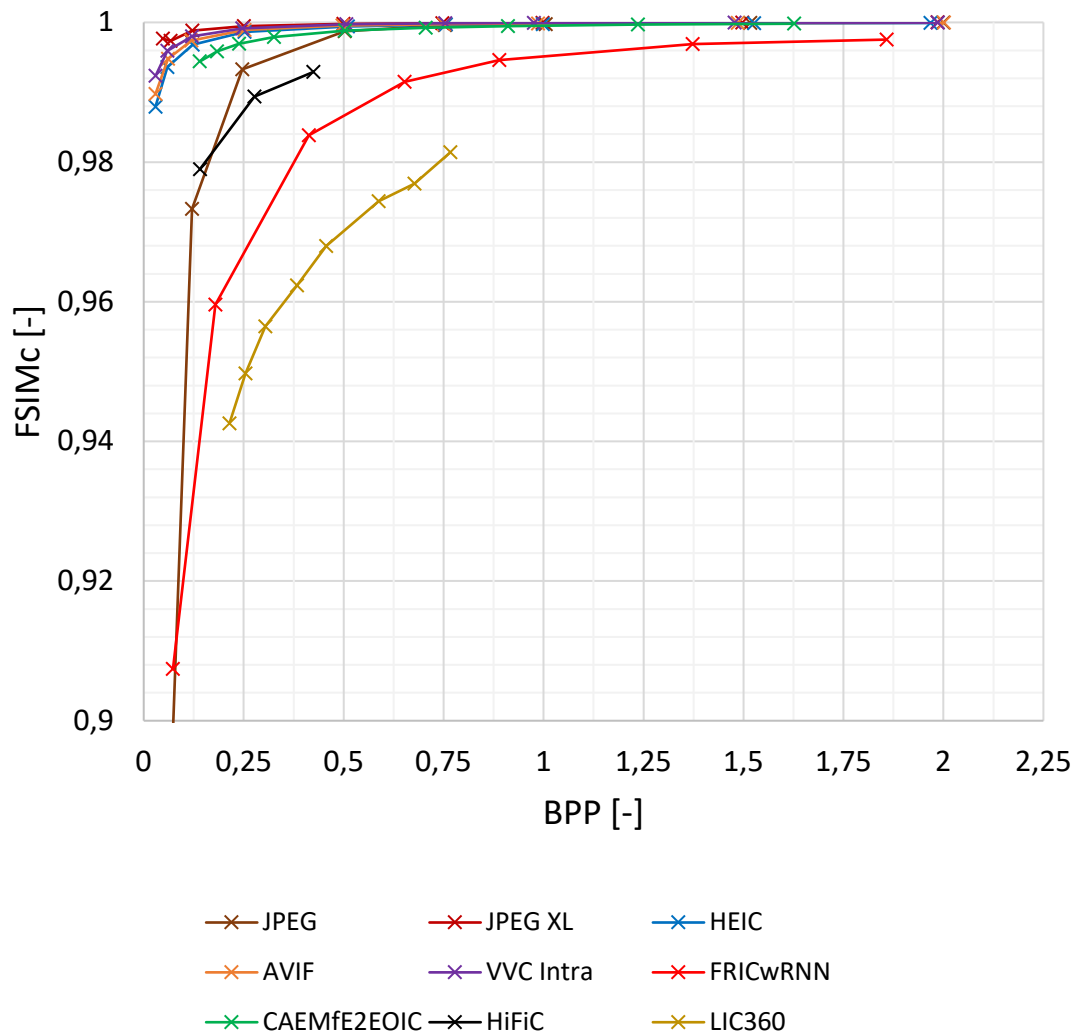
Obrázok 34 Výsledky metriky VIFp

Tabuľka 42 Porovnanie kompresnej účinnosti metódou BD-Rate, VIFp

BD-Rate [%]	JPEG	JPEG XL	HEIC	AVIF	VVC Intra	FRICwRNN	CAEMfE2EOIC	HiFiC	LIC360
JPEG	0	55	111	130	156	-66	-25	-38	-23
JPEG XL	-36	0	34	46	62	-83	-50	-64	-48
HEIC	-53	-26	0	9	22	-88	-63	-77	-63
AVIF	-56	-32	-8	0	12	-90	-67	-80	-67
VVC Intra	-61	-38	-18	-11	0	-91	-71	-84	-72
FRICwRNN	194	477	762	870	1031	0	200	117	192
CAEMfE2EOIC	34	98	170	200	246	-67	0	-42	2
HiFiC	62	177	341	412	519	-54	72	0	58
LIC360	31	91	169	202	256	-66	-2	-37	0

6.3.5 Výsledky metriky FSIMc

Na obrázku 35 a v tabuľke 43 sú zobrazené výsledky metriky FSIMc.



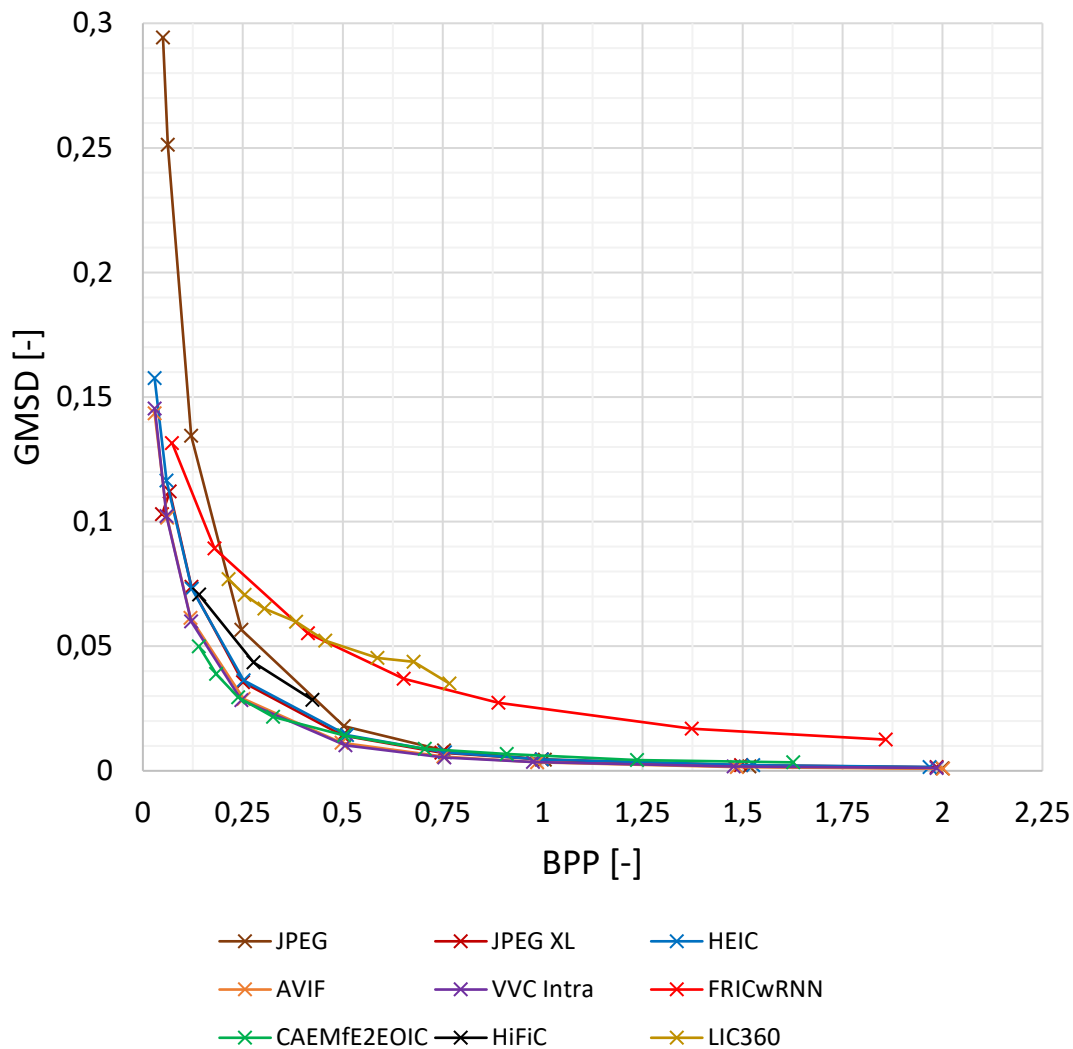
Obrázok 35 Výsledky metriky FSIMc

Tabuľka 43 Porovnanie kompresnej účinnosti metódou BD-Rate, FSIMc

BD-Rate [%]	JPEG	JPEG XL	HEIC	AVIF	VVC Intra	FRICwRNN	CAEMfE2EOIC	HiFiC	LIC360
JPEG	0	330	278	314	364	-39	41	-18	-71
JPEG XL	-79	0	-54	-44	-28	-96	-74	NaN	NaN
HEIC	-74	115	0	21	61	-93	-46	-88	NaN
AVIF	-76	79	-17	0	32	-94	-56	-90	NaN
VVC Intra	-78	38	-38	-24	0	-95	-66	-92	NaN
FRICwRNN	64	2390	1295	1518	1899	0	503	116	-48
CAEMfE2EOIC	-29	287	85	125	192	-83	0	NaN	NaN
HiFiC	22	NaN	719	876	1226	-54	NaN	0	-80
LIC360	248	NaN	NaN	NaN	NaN	94	NaN	409	0

6.3.6 Výsledky metriky GMSD

Na obrázku 36 a v tabuľke 44 sú zobrazené výsledky metriky GMSD.



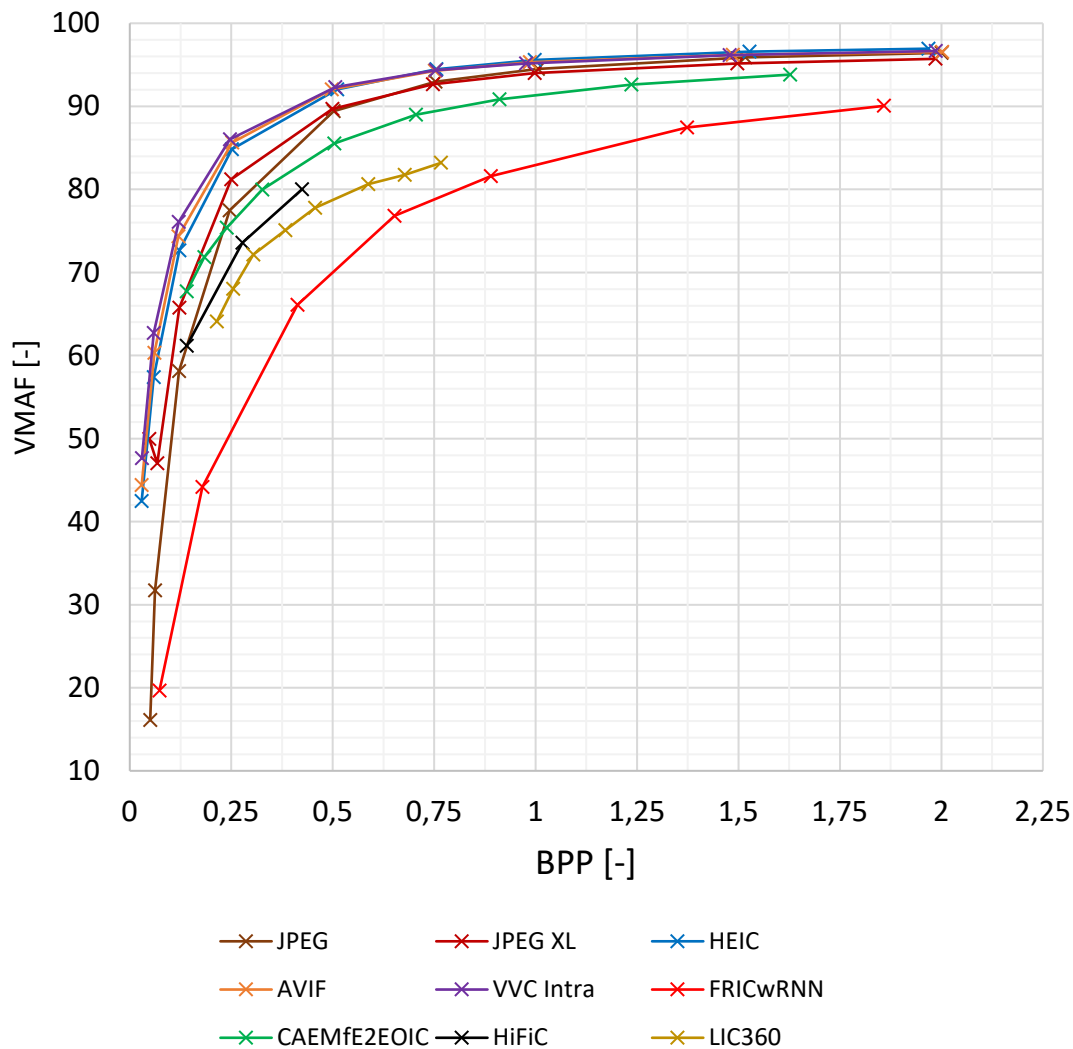
Obrázok 36 Výsledky metriky GMSD

Tabuľka 44 Porovnanie kompresnej účinnosti metódou BD-Rate, GMSD

BD-Rate [%]	JPEG	JPEG XL	HEIC	AVIF	VVC Intra	FRICwRNN	CAEMfE2EOIC	HiFiC	LIC360
JPEG	0	27	36	47	48	22	53	-58	-44
JPEG XL	-21	0	-1	11	12	-9	33	-64	-53
HEIC	-27	1	0	12	12	-7	33	-64	-53
AVIF	-32	-10	-10	0	0	-15	18	-67	-57
VVC Intra	-32	-10	-10	0	0	-15	15	-67	-57
FRICwRNN	-18	10	8	18	18	0	124	-62	-54
CAEMfE2EOIC	-35	-25	-25	-15	-13	-55	0	-81	-81
HiFiC	137	178	179	200	201	165	434	0	12
LIC360	80	112	113	131	131	117	435	-11	0

6.3.7 Výsledky metriky VMAF

Na obrázku 37 a v tabuľke 45 sú zobrazené výsledky metriky VMAF pre použité kodeky.



Obrázok 37 Výsledky metriky VMAF

Tabuľka 45 Porovnanie kompresnej účinnosti metódou BD-Rate, VMAF

BD-Rate [%]	JPEG	JPEG XL	HEIC	AVIF	VVC Intra	FRICwRNN	CAEMfE2EOIC	HiFiC	LIC360
JPEG	0	18	75	87	101	-57	-16	-20	-41
JPEG XL	-15	0	43	56	71	-69	-27	-36	-52
HEIC	-43	-30	0	9	19	-79	-44	-53	-64
AVIF	-47	-36	-8	0	10	-81	-48	-58	-67
VVC Intra	-50	-42	-16	-9	0	-83	-52	-62	-70
FRICwRNN	131	226	375	420	473	0	154	112	60
CAEMfE2EOIC	20	36	79	94	108	-61	0	-25	-41
HiFiC	25	55	111	136	163	-53	34	0	-20
LIC360	70	107	176	207	239	-37	69	26	0

6.3.8 Vyhodnotenie výsledkov porovnania jednotlivých kodekov

V prípade metriky PSNR-HVS-M sa ako najefektívnejší javil kodek VVC Intra. To isté platí aj v prípade metrík WS-PSNR, MS-SSIM, VIFp a VMAF. Kodek AVIF bol v priemere o 9% menej efektívny ako kodek VVC Intra a zároveň o 7% efektívnejší ako kodek HEIC. Pri hodnotení podľa rovnakých metrík bol kodek JPEG XL v priemere o 29% menej efektívny ako kodek HEIC, ale v prípade metriky FSIMc sa javil ako najefektívnejší. Ďalším v poradí je kodek JPEG, ktorý bol podľa očakávania najmenej efektívny z testovaných konvenčných kodekov. Ostatné konvenčné kodeky dosiahli voči kodeku JPEG o 29 – 53% vyššiu redukciu BPP pri zachovaní rovnakej kvality obrazu. Z algoritmov založených na hlbokom učení vyšiel ako najefektívnejší algoritmus CAEMfE2EOIC, avšak efektivitou kompresie nedokázal prekonať kodek JPEG, ktorý bol voči nemu v priemere o 42% efektívnejší. To však platí len pre vyššie hodnoty BPP, pretože z RD kriviek je viditeľné, že pre nižšie hodnoty BPP je efektívnejší algoritmus CAEMfE2EOIC. Podľa metriky GMSD sa algoritmus CAEMfE2EOIC javí ako najefektívnejší. Podľa metrík PSNR-HVS-M, FSIMc, GMSD a VMAF sa z dvojice algoritmov HiFiC a LIC360 javil ako efektívnejší algoritmus HiFiC. Algoritmus LIC360 bol efektívnejší v prípade metrík WS-PSNR, MS-SSIM a VIFp. V priemere bol ale algoritmus HiFiC od algoritmu LIC360 mierne efektívnejší. Podľa tabuľky BD-Rate je síce algoritmus zo všetkých najmenej efektívny, ale chyba v tomto prípade mohla byť spôsobená zlou interpoláciou kvôli nízkemu počtu referenčných bodov a údaje z RD krivky sú v tomto prípade relevantnejšie, než údaje z BD-Rate tabuľky. Ako najmenej efektívny sa podľa štyroch zo siedmych metrík a aj v priemere javil algoritmus FRICwRNN, je to však jeden z prvých algoritmov založených na hlbokom učení, rovnako ako je aj kodek JPEG jeden z prvých konvenčných kodekov na kompresiu obrázkov a preto sú takéto výsledky očakávateľné.

V prípade porovnania kompresnej účinnosti kodekov sú v tabuľkách 39 – 45 zobrazené len hodnoty BD-Rate, pretože sú používané častejšie, než hodnoty BD-PSNR. Správne hodnoty BD-PSNR boli spočítateľné len pre metriky PSNR-HVS-M, WS-PSNR a VMAF. V prípade ostatných metrík boli BD-PSNR hodnoty prakticky nulové. Kompletné BD-PSNR, ako aj BD-Rate tabuľky sú súčasťou elektronickej prílohy.

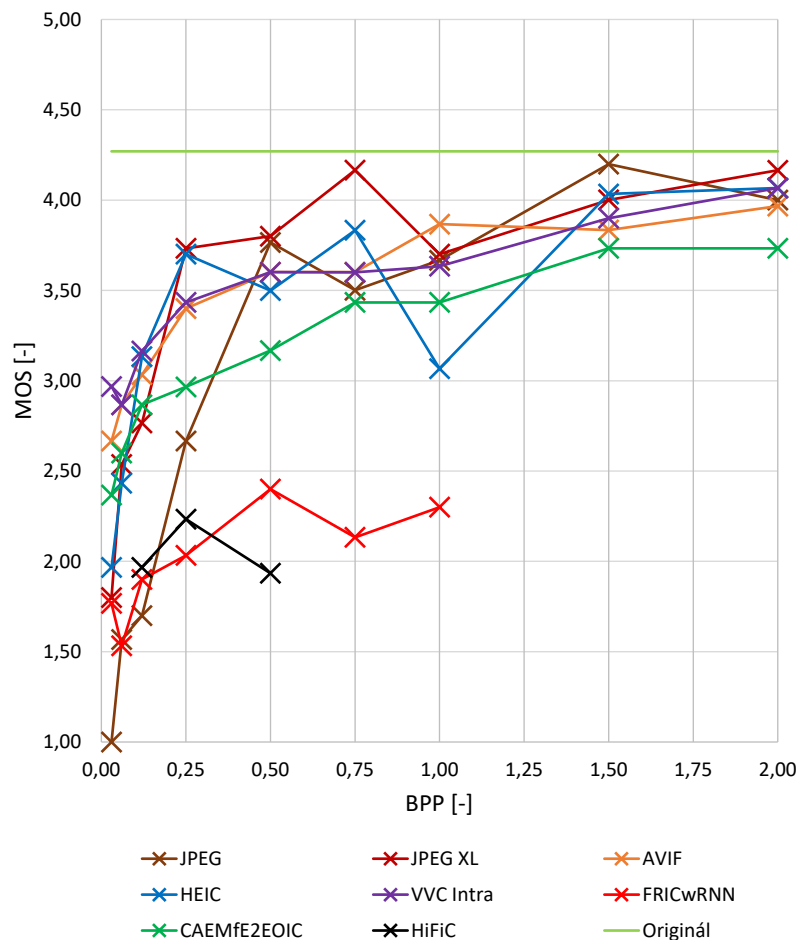
V prípade metriky FSIMc nebolo možné spočítať všetky hodnoty BD-Rate pre porovnanie algoritmov z dôvodu nemožnosti vytvoriť medzi RD krivkami integračnú plochu. Tento problém bol najviac pozorovateľný v prípade algoritmu LIC360. Tieto polia sú v tabuľke indikované hodnotou NaN.

Z tvaru RD kriviek pre kodek JPEG XL je možné pozorovať, že najnižšia hodnota BPP leží mimo trendu a má vyššiu hodnotu. To je spôsobené priemerovaním len cez päť obrázkov z datasetu miesto všetkých desiatich. Je to kvôli faktu, že túto najnižšiu hodnotu BPP nebolo pri kompresii možné dosiahnuť. Normálne by bola táto hodnota z grafu odstránená, avšak zobrazená slúži ako viditeľný dôsledok toho, že pokiaľ nie sú pri jednej hodnote BPP priemerované hodnoty kvality zo všetkých obrázkov, menej komplexné

obrázky, ktoré bolo možné skomprimovať na danú hodnotu BPP, môžu mať vo výsledku vyššiu nameranú obrazovú kvalitu, ako priemer všetkých obrázkov vo vyššom BPP. Táto skutočnosť bola pri výpočte BD-Rate zohľadnená zmenou nameranej hodnoty kvality prislúchajúcej tejto hodnote BPP podľa trendu RD krivky.

6.4 Výsledky subjektívnej metriky ACR-HR

Subjektívne metriky boli aplikované na rovnaké obrázky ako v prípade objektívnych metrick a výsledky subjektívneho testu sú zobrazené na obrázku 38. MOS skóre referenčných obrázkov bolo 4,27, čo je v grafe označené svetlozelenou farbou. Komprimované obrázky dosiahli podľa predpokladu nižšie MOS skóre, ako referenčné obrázky. Vzhľadom na nízky počet respondentov je možné vidieť vysoké kolísanie hodnôt, kde paradoxne nastáva jav, pri ktorom je obrázok komprimovaný vyššou kvalitou ohodnotený, ako menej kvalitný. Tento jav je najviac viditeľný pri kodekoch HEIC, JPEG XL a HiFiC. Z celkového pohľadu však nie sú tieto špičky až tak dôležité, ako fakt, že po bližšom preskúmaní jednotlivých kriviek a odhadu ich trendu výsledky subjektívneho testu do značnej miery odpovedajú výsledkom objektívnych metrick.



Obrázok 38 Výsledky subjektívneho hodnotenia kvality

7. PRAKTICKÝ NÁVRH FRAMEWORKU

Po vytvorení datasetu vhodných 360° obrázkov s rozlíšením 8K boli tieto obrázky podľa potreby zmenšené aj do nižších rozlíšení 2K a 4K a vytvorené výrezy s rozlíšením 1K. Následne boli všetky obrázky skomprimované piatimi konvenčnými kodekmi a štyrmi algoritmami založenými na hlbokom učení. Po kompresii boli tieto obrázky dekomprimované. Kvalita obrázkov bola zmeraná siedmymi objektívnymi metrikami a jednou subjektívnou metrikou. Kompresia a dekompresia obrázkov, výpočet objektívnych metrík, výpočet BPP a import dát do programu Excel bol vykonaný vytvoreným pomocným skriptom naprogramovaným v jazyku Python 3.9. Tento skript je uložený pod názvom `Script.py` a je súčasťou elektronickej prílohy. Okrem kvality obrázkov boli zmerané aj časy kompresie obrázkov jednotlivými kodekmi. Po získaní všetkých potrebných výsledkov bol framework naprogramovaný v jazyku Python vo verzii 3.9 dostupnom v nástroji Anaconda. Použité programovacie prostredie je Spyder vo verzii 5.2.2, taktiež dostupné v nástroji Anaconda.

7.1 Inštalácia

Pre funkčnosť frameworku je vyžadovaná správna inštalácia kodekov a prostredí obsahujúcich jednotlivé DL algoritmy. Pre algoritmus FRICwRNN musí byť vytvorené prostredie s názvom FRICwRNN, pre algoritmus CAEMfE2EOIC musí byť vytvorené prostredie s názvom CAEMfE2EOIC, pre algoritmus HiFiC musí byť vytvorené prostredie s názvom HiFiC a pre algoritmus LIC360 musí byť vytvorené prostredie s názvom LIC360. Zdrojové kódy jednotlivých algoritmov musia byť uložené v umiestnení `C:/ProgramData/Anaconda3/envs/<názov prostredia>/_Github/`. Inštalácia konvenčných kodekov, metrík a nástroja FFmpeg sa vykonáva pridaním ich `.exe` súborov do systémových premenných prostredí OS.

7.2 Knižnice

Pre funkčnosť frameworku je potrebných 10 knižníc, ktoré sú importované na začiatku skriptu. Všetky sú zobrazené v tabuľke 46. Pokiaľ niektorá bude chýbať, je nutné ju manuálne doinštalovať, inak nebude funkčnosť zaručená. V prípade použitia nástroja Anaconda bolo potrebné doinštalovať len knižnice `matlab.engine` a `pillow`, ostatné knižnice sú súčasťou nástroja Python 3.9, konkrétne jeho podverzie 3.9.13. Knižnicu `pillow` je možné doinštalovať výberom priamo v prostredí Anaconda a návod na inštaláciu knižnice `matlab.engine` možno nájsť na stránkach MathWorks [105]. Knižnice vyžadované jednotlivými DL algoritmami aj s tvorbou jednotlivých prostredí sú popísané v kapitolách 5.5 a 5.6 v podkapitolách jednotlivých algoritmov.

Tabuľka 46 Knižnice potrebné pre funkčnosť frameworku

Knižnica	Verzia
csv	python 3.9.13
glob	python 3.9.13
matlab.engine	MATLAB R2022b
os	python 3.9.13
pillow	9.2.0
shutil	python 3.9.13
subprocess	python 3.9.13
time	python 3.9.13
timeit	python 3.9.13
tkinter	python 3.9.13

7.3 Obsluha

Framework je tvorený jedným skriptom a je funkčný na zariadeniach s operačným systémom Windows 10 x64. Je uložený pod názvom Framework.py a je súčasťou elektronickej prílohy. Na iných platformách funkčnosť testovaná nebola. Ovládanie je zamerané na jednoduché ovládanie a za týmto účelom bolo vytvorené GUI s využitím knižnice tkinter.

Tlačidlom „Vybrať obrázok“ je možné vybrať obrázok, ktorý bude komprimovaný. Tento obrázok musí byť vo formáte PNG. Po zvolení obrázku sa obrázok zobrazí pod položkou „Referenčný obrázok“ a vedľa položky „Rozmery obrázku:“ bude zobrazené jeho rozlíšenie v pixeloch. Podľa výsledkov kompresie z kapitoly 5 by si mal používateľ vedieť určiť maximálne rozlíšenie obrázkov, ktoré zvládne skomprimovať jeho počítač. Maximálne ani minimálne rozlíšenie nie je zo strany frameworku limitované, maximálne odporúčané rozlíšenie je 8192×4096 (8K) pixelov, čo je dostatočné pre prezeranie 360° obrázkov v prostredí VR. Toto rozlíšenie predstavuje pomer strán 2:1, ktorý je vyžadovaný aj ERP projekciou obrázkov, na ktorú je framework cielený, avšak je možné ním komprimovať aj klasické planárne obrázky.

Pod záložkou „Kodek“ je možné vybrať, ktorými kodekmi má byť referenčný obrázok komprimovaný a dekomprimovaný. Dostupné sú konvenčné kodeky JPEG, JPEG XL, AVIF, HEIC a VVC Intra a algoritmy založené na hlbokom učení FRICwRNN, CAEMfE2EOIC, HiFiC a LIC360. Všetky konvenčné kodeky sú nastavené na maximálnu komplexnosť kompresie a pri DL algoritmoch, pri ktorých to bolo možné, je vybratá optimalizácia pre metriky založené na metrike SSIM.

Pod záložkou „Kvalita/QP“ je možné navoliť kvalitu, respektíve kvantizačné parametre kompresie dostupné pre dané nastavenia kodekov.

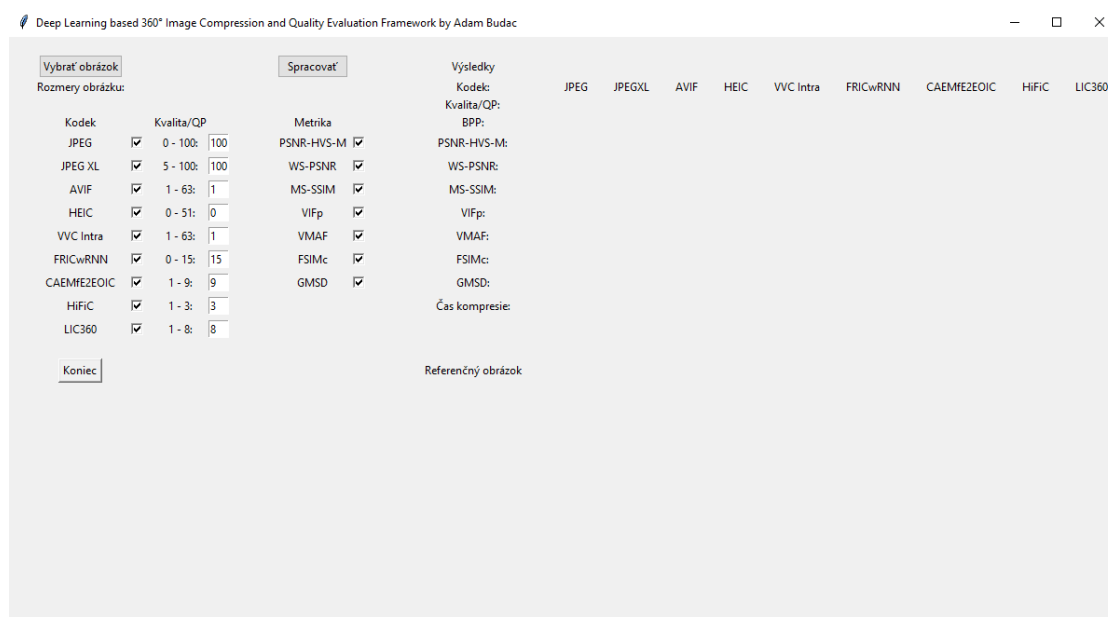
Pod záložkou „Metrika“ sú na výber objektívne metriky, ktorými bude vyhodnotená kvalita komprimovaných obrázkov. Predvolene sú vybraté všetky kodeky aj metriky a kvalita kompresie je nastavená na maximum. Aby bol framework funkčný, všetky

použité kodeky a metriky musia byť nainštalované na užívateľskom počítači podľa návodu. Framework bude fungovať aj v prípade, že niektoré kodeky a metriky nebudú nainštalované, no tieto nenainštalované kodeky a metriky je treba odškrtnúť v príslušných zaškrťovacích poliach, aby neboli použité.

Tlačidlom „Spracovať“ je následne vybraný obrázok spracovaný nástrojom FFmpeg, ktorý je pre funkčnosť frameworku vyžadovaný. Odkazy na všetky tieto softwarové nástroje sú uvedené v použitej literatúre. Ďalej je v rámci spracovania obrázkov skomprimovaný a dekomprimovaný vybranými kodekmi a vybranými objektívnymi metrikami je zmeraná jeho kvalita. Tiež je vypočítaná hodnota BPP komprimovaného obrázku pre každý použitý kodek a zmeraná je aj čas kompresie jednotlivými zvolenými kodekmi. Zároveň je v ďalšom okne zobrazený príkazový riadok s práve vykonávanými príkazmi. V prípade spúšťania skriptu z vývojového prostredia, akým je napríklad už zmieneny Spyder, okno s príkazovým riadkom zobrazené nebude.

Po dokončení spracovania budú v časti „Výsledky“ zobrazené výsledky spracovania. Sem patrí kvalita nastavená jednotlivým kodekom, vypočítané hodnoty BPP, výsledky objektívnych metrick a zmeraný čas kompresie pre každý použitý kodek. Po preskúmaní výsledkov je možné framework ukončiť tlačidlom „Koniec“.

Na obrázku 39 je zobrazený spustený framework a na obrázku 40 je zobrazený framework po spracovaní referenčného obrázku.



Obrázok 39 Spustený framework

Deep Learning based 360° Image Compression and Quality Evaluation Framework by Adam Budac


Vybrat' obrázok: inc.png Spracovat'

Rozmery obrázku: 512 x 256

Kodek	Kvalita/QP	Metrika	Výsledky
JPEG	<input checked="" type="checkbox"/> 0 - 100: [80]	PSNR-HVS-M <input checked="" type="checkbox"/>	Kodek: JPEG 80 70 20 15 20
JPEG XL	<input checked="" type="checkbox"/> 5 - 100: [70]	WS-PSNR <input checked="" type="checkbox"/>	Kvalita/QP: 1.747 0.875 1.181 2.201 2.469
AVIF	<input checked="" type="checkbox"/> 1 - 63: [20]	MS-SSIM <input checked="" type="checkbox"/>	BPP: 53.844 43.277 51.049 58.010 63.252
HEIC	<input checked="" type="checkbox"/> 0 - 51: [15]	VIFp <input checked="" type="checkbox"/>	PSNR-HVS-M: 41.975 38.371 43.982 48.785 51.299
VVC Intra	<input checked="" type="checkbox"/> 1 - 63: [20]	VMAF <input checked="" type="checkbox"/>	MS-SSIM: 0.9986 0.9962 0.9987 0.9995 0.9997
FRICwRNN	<input checked="" type="checkbox"/> 0 - 15: [13]	FSIMc <input checked="" type="checkbox"/>	VIFp: 0.8550 0.7711 0.8860 0.9480 0.9684
CAEMFE2EOIC	<input checked="" type="checkbox"/> 1 - 9: [8]	GMSD <input checked="" type="checkbox"/>	VMAF: 95.579 90.131 94.911 96.865 97.063
HIFIC	<input checked="" type="checkbox"/> 1 - 3: [3]		FSIMc: 0.9904 0.9785 0.9928 0.9965 0.9986
LIC360	<input checked="" type="checkbox"/> 1 - 8: [6]		GMSD: 0.0020 0.0110 0.0036 0.0015 0.0008
			Čas kompresie: 00:00:00 00:00:00 00:00:06 00:00:01 00:00:14 00:00:27 00:00:44 00:00:16 00:00:11

Koniec

Referenčný obrázok



Obrázok 40 Framework po spracovaní obrázku a výpočte výsledkov

8. ZÁVER

V teoretickej časti práce bol podaný prehľad o projekciách a deformáciách 360° obrazu. Ďalej boli popísané konvenčné kodeky určené pre kompresiu obrazu. V časti venujúcej sa umelej inteligencii boli vysvetlené súvislosti medzi strojovým učením, neurónovými sieťami a hlbokým učením, architektúry jednotlivých sietí a možnosti ich využitia pre kompresiu 360° obrazu. Posledná teoretická kapitola je venovaná objektívnym a subjektívnym metrikám vhodným pre meranie kvality 360° obrazu.

V praktickej časti práce je vytvorený dataset 10 obrázkov s 360° uhlom záberu. Tieto obrázky sú v rozlíšení 8K. Zároveň bol spočítaný aj priestorový index týchto obrázkov. Tieto obrázky boli podľa potreby zmenšené aj do nižších rozlíšení 2K a 4K a vytvorené výrezy s rozlíšením 1K. Následne boli všetky obrázky skomprimované piatimi konvenčnými kodekmi a štyrmi algoritmami založenými na hlbokom učení. Testované konvenčné kodeky boli JPEG, JPEG XL, HEIC, AVIF a VVC Intra. Testované algoritmy hlbokého učenia boli Full Resolution Image Compression with Recurrent Neural Networks (FRICwRNN), Context-adaptive Entropy Model for End-to-end Optimized Image Compression (CAEMfE2EOIC), High-Fidelity Generative Image Compression (HiFiC) a End-to-End Optimized 360° Image Compression (LIC360). Rozlíšenie obrázkov, ktoré bolo možné komprimovať na použitom HW je v prípade konvenčných kodekov pôvodné 8K. Pri použití algoritmov hlbokého učenia to bolo 4K v prípade algoritmu CAEMfE2EOIC a 2K pri použití algoritmov FRICwRNN a HiFiC. Kompresia vo vyššom rozlíšení nebola možná z dôvodu vysokých nárokov algoritmov na pamäte RAM a VRAM. Algoritmus LIC360 komprimoval len výrezy z obrázkov v 1K rozlíšení, kde tieto výrezy predstavovali plochu videnu respondentom v prostredí VR. Algoritmy FRICwRNN, HiFiC a LIC360 využili aj hardwarovú akceleráciu GPU na urýchlenie kompresie. V rýchlosti kompresie bolo poradie kodekov zoradených od najrýchlejšieho nasledovné: LIC360, JPEG, HiFiC, FRICwRNN, AVIF, HEIC, JPEG XL, CAEMfE2EOIC a VVC Intra. Po jednoduchom prepočítaní času kompresie algoritmov hlbokého učenia do rozlíšenia 8K by bolo poradie kodekov zmenené a to: JPEG, AVIF, HEIC, JPEG XL, LIC360, HiFiC, VVC Intra, CAEMfE2EOIC a FRICwRNN. Dekompresia konvenčnými kodekmi bola takmer okamžitá, avšak dekompresia algoritmami hlbokého učenia bola vplyvom architektúry neurónových sietí rovnako dlhá, ako kompresia. Po následnej dekompresii bola zameraná kvalita obrázkov siedmimi objektívnymi metrikami. Tieto metriky sú PSNR-HVS-M, WS-PSNR, MS-SSIM, VIFp, FSIMc, GMSD a VMAF. Výsledky objektívnych metrik pre použité kodeky zoradené od najefektívnejšieho sú nasledovné: VVC Intra, AVIF, HEIC, JPEG XL, JPEG pre vyššie BPP, CAEMfE2EOIC pre nižšie BPP, HiFiC, LIC360 a FRICwRNN. Tieto dosiahnuté výsledky boli prezentované aj na konferenciách MAREW 2023 a STUDENT EEICT 2023.

Pre nenáročné aplikácie a aplikácie s nízkym výpočtovým výkonom a energetickou spotrebou sa kodek JPEG javí ako najvhodnejšia voľba. Pre aplikácie vyžadujúce vysokú kvalitu obrazu s možnosťou vyšších nárokov na spotrebu energie, výpočtový výkon a bez najvyšších požiadaviek na výslednú veľkosť je podľa testu vhodná voľba kodek HEIC a o niečo vhodnejšia voľba kodek AVIF. Ich výhoda je aj možnosť použitia HW dekodérov pri dekompresii. Pri potrebe čo najviac znížiť veľkosť výsledného obrázku bez ohľadu na spotrebu energie či rýchlosť kompresie sa javí ako najvhodnejšia voľba kodek VVC Intra. Kodek JPEG XL je vhodný na archiváciu obrázkov aj pre jeho možnosť ďalej skomprimovať komprimované JPEG obrázky bez ďalšej straty kvality a pri použití nižšej komplexnosti kompresie by mohol byť rýchlejší ako HEIC a AVIF. Jeho výhoda je aj pre možnosť použitia na kompresiu obrázkov s mnohonásobne vyšším rozlíšením, ako testované 8K rozlíšenie, čím sú obmedzené kodeky odvodené z kodekov určených pre kompresiu videosekvencií. Algoritmy založené na hlbokom učení nedosiahli v testoch na výsledky konvenčných kodekov, avšak s trendom rýchlosti výskumu a vývoja umelej inteligencie a algoritmov strojového a hlbokého učenia bude zaujímavé sledovať napredovanie obraz komprimujúcich algoritmov založených na princípoch JPEG AI.

Na overenie výsledkov objektívnych metrick bol vykonaný aj subjektívny test na troch respondentoch pomocou metriky ACR-HR v prostredí VR za použitia headsetu Oculus Rift S. Títo respondenti sú odborníkmi v oblasti obrazovej techniky. Výsledky subjektívneho testu odpovedali objektívnemu hodnoteniu.

Po získaní všetkých výsledkov bol v programovacom jazyku Python vytvorený framework, ktorým je možné komprimovať 360° obrázky prostredníctvom konvenčných kodekov a algoritmov strojového učenia, počítať kvalitu komprimovaných obrázkov objektívnymi metrikami, počítať hodnotu BPP a merať čas kompresie jednotlivými kodekmi a to všetko plne automatizovane v jednoduchom a intuitívnom grafickom prostredí.

V budúcej práci budú preskúmané ďalšie možnosti a zlepšenia kompresie obrazu pomocou hlbokého učenia, kompresia videosekvencií prostredníctvom algoritmov strojového a hlbokého učenia, ako aj rozdiely v objektívnej a subjektívnej kvalite medzi komprimovaným obrazovým materiálom v štandardnom a vysokom dynamickom rozsahu v širokom farebnom priestore, prípadne aj v 3D či v 360° uhle záberu.

LITERATÚRA

- [1] BUDÁČ, Adam. Laboratorní pracoviště pro virtuální realitu [online]. Brno, 2020 [cit. 2022-10-09]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/128340>. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Vedoucí práce Jan Kufa.
- [2] VR headset Valve Index. *IGN* [online]. San Francisco (CA): IGN Entertainment, c1996-2022, 11 Oct 2022 [cit. 2022-10-09]. Dostupné z: https://assets1.ignimgs.com/2019/06/26/valveindex-1280-1561582937647_160w.jpg?width=1280
- [3] *Gigapixel.com* [online]. Port Moody (BC), c2022 [cit. 2022-10-09]. Dostupné z: <https://www.gigapixel.com/>
- [4] *Gigapan* [online]. Portland (OR), c2013 [cit. 2022-10-09]. Dostupné z: <http://gigapan.com/>
- [5] *360Cities* [online]. Zeewolde (NL), c2014-2021 [cit. 2022-10-09]. Dostupné z: <https://www.360cities.net/>
- [6] *Google mapy* [online]. Mountain View (CA), c2022 [cit. 2022-10-09]. Dostupné z: <https://www.google.com/maps>
- [7] Profesionálna 360° kamera. In: My360 [online]. 2017 [cit. 2022-10-09]. Dostupné z: https://my360propertyvirtualtours.com/wp-content/uploads/2017/03/iris360_product_rev8.png
- [8] SIMKA, Marek, Jan KUFA a Ladislav POLAK. Picture Quality of 360° Images Compressed by Emerging Compression Algorithms. In: 2022 32nd International Conference Radioelektronika (RADIOELEKTRONIKA) [online]. IEEE, 2022, 2022-4-21, s. 1-4 [cit. 2022-10-09]. ISBN 978-1-7281-8686-3. Dostupné z: doi:10.1109/RADIOELEKTRONIKA54537.2022.9764941
- [9] XU, Mai, Chen LI, Shanyi ZHANG a Patrick Le CALLET. State-of-the-Art in 360° Video/Image Processing: Perception, Assessment and Compression. *IEEE Journal of Selected Topics in Signal Processing* [online]. 2020, 14(1), 5-26 [cit. 2022-10-09]. ISSN 1932-4553. Dostupné z: doi:10.1109/JSTSP.2020.2966864
- [10] Mapa dôrazu. *V-SENSE* [online]. Dublin (IE): Trinity College Dublin, c2018, 10 May 2018 [cit. 2022-10-25]. Dostupné z: <https://v-sense.scss.tcd.ie/research/3dof/salnet360-saliency-maps-for-omni-directional-images-with-cnn/>
- [11] CHEN, Zhenzhong, Yiming LI a Yingxue ZHANG. Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation. *Signal Processing* [online]. ScienceDirect, 2018, 10 May 2018, 146, 66-78 [cit. 2022-10-25]. ISSN 01651684. Dostupné z: doi:10.1016/j.sigpro.2018.01.004
- [12] ERP projekcia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022, 6 September 2022 [cit. 2022-10-25]. Dostupné z:

- https://upload.wikimedia.org/wikipedia/commons/8/83/Equirectangular_projection_SW.jpg
- [13] EAP projekcia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022, 5 April 2022 [cit. 2022-10-25]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/7/76/Cylindrical_equal-area_projection_SW.jpg
- [14] CMP projekcia. *Stack Exchange: Geographic Information Systems Stack Exchange* [online]. New York (NY), 2021, 27 Mar 2021 [cit. 2022-10-25]. Dostupné z: <https://i.stack.imgur.com/ulvyP.png>
- [15] CPP projekcia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022, 8 September 2022 [cit. 2022-10-25]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/9/9c/Craster_parabolic_projection_SW.jpg
- [16] JPEG zdrojové kódovanie. *World Wide Web Consortium (W3C)* [online]. Cambridge (MA): World Wide Web Consortium, 2022, 09 1992 [cit. 2022-10-25]. Dostupné z: <https://www.w3.org/Graphics/JPEG/itu-t81.pdf>
- [17] Farebné podvzorkovanie. In: RAVe [PUBS] [online]. rAVe Publications, 2016 [cit. 2022-10-25]. Dostupné z: https://2ley7l42nt9s3jvzio2zneqa-78.wpengine.netdna-ssl.com/wpcontent/uploads/2016/02/Common_chroma_subsampling_ratios-0216.jpg
- [18] HANUS, Stanislav. *JPEG kodér a dekodér* [online]. Brno (CZ), 2020, 2020 [cit. 2022-10-25]. Dostupné z: <https://www.vut.cz/>
- [19] HEVC, HEIC zdrojové kódovanie. ITU: Committed to connecting the world [online]. Ženeva (CH): ISO/IEC, 2008, 29.11.2019 [cit. 2021-03-01]. Dostupné z: <https://www.itu.int/rec/T-REC-H.265-202108-I/en>
- [20] HEIC stromová štruktúra. In: ResearchGate [online]. 2014 [cit. 2022-10-25]. Dostupné z: https://www.researchgate.net/profile/Detlev_Marpe/publication/300315241/figure/fig5/AS:357589742309380@1462267446547/Example-for-the-partitioning-of-a64-64-luma-coding-tree-block-black-into-coding-blocks.png
- [21] AV1, AVIF zdrojové kódovanie. Alliance for Open Media: Next Generation, Open-Source Digital Media Technology for Everyone [online]. Wakefield (MA), 2015, 18.1.2019 [cit. 2021-03-01]. Dostupné z: <http://aomedia.org/av1/specification/>
- [22] VP9, AV1, AVIF vyčítavanie koeficientov. In: Doom9's Forum [online]. 2013 [cit. 2022-10-25]. Dostupné z: <https://i.imgur.com/H41WZMK.png>
- [23] JPEG XL zdrojové kódovanie. *JPEG XL* [online]. Ženeva (CH): ITU-T, 2022 [cit. 2022-10-25]. Dostupné z: https://docs.google.com/presentation/d/1LlmUR0Uoh4dgT3DjanLjhlXrk_5W2nJBDqDAMBhe8v8/edit#slide=id.g821b9c186b_0_34

- [24] VVC, VVC Intra: Versatile video coding. *ITU: Committed to connecting the world* [online]. Ženeva (CH): ITU, c2022, 29 04 2022 [cit. 2022-12-04]. Dostupné z: <https://www.itu.int/rec/T-REC-H.266-202204-I/en>
- [25] ŠAFÁŘ, Daniel. Efektivní kódování obrazu založené na učení. *ČVUT DSpace* [online]. Praha (CZ), c2016, Květen 2021 [cit. 2022-11-23]. Dostupné z: <https://dspace.cvut.cz/handle/10467/94804>
- [26] GLÓS, Kristián. *Hluboké učení AI v herních prostředích: Deep Learning AI in Game Environments* [online]. Brno (CZ), 2021 [cit. 2022-11-30]. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/136594>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Tomáš Polášek.
- [27] ANGRA, Sheena a Sachin AHUJA. Machine learning and its applications: A review. *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)* [online]. Ženeva (CH): IEEE, 2017, 2017, 57-60 [cit. 2022-11-30]. ISBN 978-1-5090-6400-7. Dostupné z: doi:10.1109/ICBDACI.2017.8070809
- [28] INTRO TO AI #1: ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, NEURAL NETWORKS AND DEEP LEARNING: WHAT ARE THE DIFFERENCES?. *Aiso-lab: Deep learning for your business* [online]. Köln (DE), c2017, 14. SEPTEMBER 2017 [cit. 2022-11-23]. Dostupné z: <https://aiso-lab.com/intro-to-ai-1-artificial-intelligence-machine-learning-neural-networks-and-deep-learning-what-are-the-differences/>
- [29] MURÁŇ, Juraj. Algoritmy strojového učenia I. – Učenie s učiteľom. *UMELÁ INTELIGENCIA.SK* [online]. c2022, 5. júna 2019 [cit. 2022-11-23]. Dostupné z: <https://umelainteligencia.sk/algoritmy-strojoveho-ucenia/>
- [30] Data collection and pre-processing techniques. *Qualcomm Developer Network* [online]. San Diego (CA), c2022 [cit. 2022-11-23]. Dostupné z: <https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk/learning-resources/ai-ml-android-neural-processing/data-collection-pre-processing>
- [31] MURÁŇ, Juraj. Algoritmy strojového učenia II. – Učenie bez učiteľa. *UMELÁ INTELIGENCIA.SK* [online]. c2022, 14. júna 2019 [cit. 2022-11-23]. Dostupné z: <https://umelainteligencia.sk/algoritmy-strojoveho-ucenia/>
- [32] MURÁŇ, Juraj. Algoritmy strojového učenia III. – Učenie formou odmeňovania. *UMELÁ INTELIGENCIA.SK* [online]. c2022, 14. júna 2019 [cit. 2022-11-23]. Dostupné z: <https://umelainteligencia.sk/algoritmy-strojoveho-ucenia/>
- [33] Single neuron in neural network using c++. *Stack Overflow* [online]. New York (NY), c2022, 9 Aug 2016 [cit. 2022-11-23]. Dostupné z: <https://stackoverflow.com/questions/38850538/single-neuron-in-neural-network-using-c>

- [34] SCHULZ, Hannes a Sven BEHNKE. Deep Learning: Layer-Wise Learning of Feature Hierarchies. *KI - Künstliche Intelligenz* [online]. Berlín (DE): Springer, 2012, 17 May 2012, (26), 357-363 [cit. 2022-11-30]. ISBN 978-1-5090-6400-7. Dostupné z: doi:10.1007/s13218-012-0198-z
- [35] HUSSAIN, Farhan a Jechang JEONG. Exploiting deep neural networks for digital image compression. *2015 2nd World Symposium on Web Applications and Networking (WSWAN)* [online]. Ženeva (CH): IEEE, 2015, 20 August 2015, 1-6 [cit. 2022-11-30]. ISBN 978-1-4799-8172-4. Dostupné z: doi:10.1109/WSWAN.2015.7210294
- [36] JADON, Shruti. Introduction to Different Activation Functions for Deep Learning. *Medium: Where good ideas find you* [online]. San Francisco (CA), 15 August 2012n. 1., 16 Mar 2018 [cit. 2022-12-14]. Dostupné z: <https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092>
- [37] DESHPANDE, Adit. A Beginner's Guide To Understanding Convolutional Neural Networks Part 2. *GitHub* [online]. San Francisco (CA), c2022, 29 July 2016 [cit. 2022-12-14]. Dostupné z: <https://adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/>
- [38] NEUTELINGS, Izaak. Neural networks. *TikZ.net* [online]. c2021, 9 Aug 2016 [cit. 2022-11-23]. Dostupné z: https://tikz.net/neural_networks/
- [39] SAK, Hasim, Andrew SENIOR a Francoise BEAUFAYS. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. *INTERSPEECH* [online]. Mountain View (CA): IEEE, 2014, 2014, 338-342 [cit. 2022-11-30]. ISBN 978-1-4799-8172-4. Dostupné z: <https://storage.googleapis.com/pub-tools-public-publication-data/pdf/43905.pdf>
- [40] Recurrent Neural Network (RNN): Rekurentná neurónová sieť. *Artificial Intelligence Wiki: AI Wiki* [online]. 2020, 2019 [cit. 2022-12-21]. Dostupné z: <https://machine-learning.paperspace.com/wiki/recurrent-neural-network-rnn>
- [41] HUSZAR, Ferenc, Lucas THEIS, Wenzhe SHI a Andrew CUNNINGHAM. Lossy Image Compression with Compressive Autoencoders. *International Conference on Learning Representations (ICLR 2017)* [online]. Cambridge (UK): Cambridge University Press, 2017, 2017 [cit. 2022-11-30]. ISBN 978-1-4799-8172-4. Dostupné z: doi:10.17863/CAM.51995
- [42] JIANG, Feng, Wen TAO, Shaohui LIU, Jie REN, Xun GUO a Debin ZHAO. An End-to-End Compression Framework Based on Convolutional Neural Networks. *IEEE Transactions on Circuits and Systems for Video Technology* [online]. Ženeva (CH): IEEE, 2018, 1 August 2017, 28(10), 3007-3018 [cit. 2022-11-30]. ISSN 1051-8215. Dostupné z: doi:10.1109/TCSVT.2017.2734838
- [43] GOODFELLOW, Ian J., Jean POUGET-ABADIE, Mehdi MIRZA, Bing XU, David WARDE-FARLEY, Sherjil OZAIR, Aaron COURVILLE a Yoshua BENGIO. *Generative Adversarial Nets* [online]. Montreal (CA): arXiv, 10 Jun

- 2014, v1 [cit. 2022-12-21]. Dostupné z:
doi:<https://doi.org/10.48550/arXiv.1406.2661>
- [44] Overview of GAN Structure. *Google Developers* [online]. Mountain View (CA), 18 07 2022 [cit. 2022-12-21]. Dostupné z:
https://developers.google.com/machine-learning/gan/gan_structure
- [45] ASCENSO, João a Evgeniy UPENIK, ed. White Paper on JPEG AI Scope and Framework. *JPEG* [online]. Ženeva (CH), 2021, January 2021 [cit. 2022-11-30]. Dostupné z: <https://ds.jpeg.org/whitepapers/jpeg-ai-white-paper.pdf>
- [46] JPEG AI Common Training and Test Conditions. *JPEG* [online]. Ženeva (CH), 2022, 21 January 2022 [cit. 2022-11-30]. Dostupné z:
https://ds.jpeg.org/documents/jpegai/wg1n100106-094-ICQ-JPEG_AI_Common_Training_and_Test_Conditions.pdf
- [47] ASCENSO, Joao, Pinar AKYAZI, Fernando PEREIRA a Touradj EBRAHIMI. Learning-based image coding: early solutions reviewing and subjective quality evaluation. *Optics, Photonics and Digital Technologies for Imaging Applications VI* [online]. online (FR): SPIE Digital Library, 2020, 1 April 2020, 11353(113530S) [cit. 2022-12-05]. Dostupné z:
doi:doi.org/10.1117/12.2555368
- [48] MAHMOUDIAN BIDGOLI, Navid, Roberto G. DE A. AZEVEDO, Thomas MAUGEY, Aline ROUMY a Pascal FROSSARD. OSLO: On-the-Sphere Learning for Omnidirectional Images and Its Application to 360-Degree Image Compression. *IEEE Transactions on Image Processing* [online]. Ženeva (CH): IEEE, 2022, 2 September 2022, 31, 5813-5827 [cit. 2022-12-09]. ISSN 1057-7149. Dostupné z: doi:[10.1109/TIP.2022.3202357](https://doi.org/10.1109/TIP.2022.3202357)
- [49] TODERICI, George, Damien VINCENT, Nick JOHNSTON, Sung Jin HWANG, David MINNEN, Joel SHOR a Michele COVELL. Full Resolution Image Compression with Recurrent Neural Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* [online]. Ženeva (CH): IEEE, 2017, 26 July 2017, 5435-5443 [cit. 2022-12-21]. ISBN 978-1-5386-0457-1. Dostupné z: doi:[10.1109/CVPR.2017.577](https://doi.org/10.1109/CVPR.2017.577)
- [50] LEE, Jooyoung, Seunghyun CHO a Seung-Kwon BEACK. Context-adaptive Entropy Model for End-to-end Optimized Image Compression. *Image and Video Processing (eess.IV): ICLR 2019* [online]. Montreal (CA): arXiv, 6 May 2019 [cit. 2022-12-21]. Dostupné z: doi:<https://doi.org/10.48550/arXiv.1809.10452>
- [51] MENTZER, Fabian, George TODERICI, Michael TSCHANNEN a Eirikur AGUSTSSON. High-Fidelity Generative Image Compression. *NeurIPS 2020: Image and Video Processing (eess.IV)* [online]. Montreal (CA): arXiv, 23 Oct 2020 [cit. 2022-12-21]. Dostupné z:
doi:<https://doi.org/10.48550/arXiv.2006.09965>
- [52] LI, Mu, Jinxing LI, Shuhang GU, Feng WU a David ZHANG. End-to-End Optimized 360° Image Compression. *IEEE Transactions on Image Processing* [online]. Ženeva (CH): IEEE, c2022, 27 September 2022, 31, 6267-

- 6281 [cit. 2022-12-09]. ISSN 1057-7149. Dostupné z:
doi:10.1109/TIP.2022.3208429
- [53] SOHAG, Saeed Ahmed, Kabirul ISLAM a Baharul ISLAM. Příklad metriky PSNR: A Novel Approach for Image Steganography Using Dynamic Substitution and Secret Key. *American Journal of Engineering Research (AJER)* [online]. ResearchGate, 2013, September 2013, 02(09), 118-126 [cit. 2022-11-30]. ISSN 2320-0847. Dostupné z:
https://www.researchgate.net/publication/268807675_A_Novel_Approach_for_Image_Steganography_Using_Dynamic_Substitution_and_Secret_Key
- [54] Metrika PSNR-HVS. In: *ResearchGate* [online]. Berlín (DE): ResearchGate, 05 Jul 2022n. 1., January 2006 [cit. 2022-10-25]. Dostupné z:
https://www.researchgate.net/publication/251229783_A_NEW_FULL-REFERENCE_QUALITY_METRICS_BASED_ON_HVS
- [55] Metrika PSNR-HVS-M. In: *ResearchGate* [online]. Berlín (DE): ResearchGate, 05 Jul 2022n. 1., January 2007 [cit. 2022-10-25]. Dostupné z:
https://www.researchgate.net/publication/242309240_On_between-coefficient_contrast_masking_of_DCT_basis_functions
- [56] CHANDLER, D.M. a S.S. HEMAMI. Metrika VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images. *IEEE Transactions on Image Processing* [online]. IEEE, 2007, 13 August 2007, 16(9), 2284-2298 [cit. 2022-11-05]. ISSN 1057-7149. Dostupné z: doi:10.1109/TIP.2007.901820
- [57] DE A. AZEVEDO, Roberto G., Neil BIRKBECK, Ivan JANATRA, Balu ADSUMILLI a Pascal FROSSARD. Metriky S-PSNR, WS-PSNR a CPP-PSNR: A Viewport-Driven Multi-Metric Fusion Approach for 360-Degree Video Quality Assessment. *2020 IEEE International Conference on Multimedia and Expo (ICME)* [online]. IEEE, 2020, 09 June 2020, 1-6 [cit. 2022-11-05]. ISBN 978-1-7281-1331-9. ISSN 1945-788X. Dostupné z:
doi:10.1109/ICME46284.2020.9102936
- [58] CHANNAPPAYYA, Sumohana S., Alan C. BOVIK a Robert W. HEATH. Metrika SSIM: A Linear Estimator Optimized for the Structural Similarity Index and its Application to Image Denoising. *2006 International Conference on Image Processing* [online]. IEEE, 2006, 20 February 2007, 2637-2640 [cit. 2022-11-05]. ISBN 1-4244-0480-0. ISSN 2381-8549. Dostupné z:
doi:10.1109/ICIP.2006.313051
- [59] *Příklad metriky SSIM: Signal Processing Library 1.0 documentation* [online]. In: . c2013 [cit. 2022-11-30]. Dostupné z: <https://mubeta06.github.io/python/sp//ssim-1.png>
- [60] WANG, Z., E.P. SIMONCELLI a A.C. BOVIK. Metrika MS-SSIM: Multiscale structural similarity for image quality assessment. *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003* [online]. IEEE, 2003, 04 May 2004, 1398-1402 [cit. 2022-11-05]. ISBN 0-7803-8104-1. Dostupné z:
doi:10.1109/ACSSC.2003.1292216

- [61] SHEIKH, H.R. a A.C. BOVIK. Metrika VIFp: Image information and visual quality. *IEEE Transactions on Image Processing* [online]. IEEE, 2006, 16 January 2006, 15(2), 430-444 [cit. 2022-11-05]. ISSN 1057-7149. Dostupné z: doi:10.1109/TIP.2005.859378
- [62] LIN ZHANG, LEI ZHANG, XUANQIN MOU a D. ZHANG. Metrika FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing* [online]. IEEE, 2011, 31 January 2011, 20(8), 2378-2386 [cit. 2022-11-05]. ISSN 1057-7149. Dostupné z: doi:10.1109/TIP.2011.2109730
- [63] LI, Fangfang, Vladimir LUKIN, Krzysztof OKARMA a Yanjun FU. FSIM a FSIMc hodnotenie kvality obrázkov: Providing a Desired Quality of BPG Compressed Images for FSIM Metric. *2021 IEEE 3rd International Conference on Advanced Trends in Information Theory (ATIT)* [online]. IEEE, 2021, 20 January 2022, 10-14 [cit. 2022-11-05]. ISBN 978-1-6654-3847-6. Dostupné z: doi:10.1109/ATIT54053.2021.9678522
- [64] XUE, Wufeng, Lei ZHANG, Xuanqin MOU a Alan C. BOVIK. GMSD: A Highly Efficient Perceptual Image Quality Index. *IEEE Transactions on Image Processing* [online]. IEEE, 2014, 03 December 2013, 23(2), 684-695 [cit. 2022-11-05]. ISSN 1057-7149. Dostupné z: doi:10.1109/TIP.2013.2293423
- [65] BAMPIS, Christos G., Zhi LI a Alan C. BOVIK. VMAF: SpatioTemporal Feature Integration and Model Fusion for Full Reference Video Quality Assessment. *IEEE Transactions on Circuits and Systems for Video Technology* [online]. IEEE, 2019, April 2018, 29(8), 2256-2270 [cit. 2022-11-05]. ISSN 1051-8215. Dostupné z: doi:10.1109/TCSVT.2018.2868262
- [66] ORDUNA, Marta, Cesar DIAZ, Lara MUNOZ, Pablo PEREZ, Ignacio BENITO a Narciso GARCIA. VMAF na 360° obraze: Video Multimethod Assessment Fusion (VMAF) on 360VR Contents. *IEEE Transactions on Consumer Electronics* [online]. IEEE, 2020, 05 December 2019, 66(1), 22-31 [cit. 2022-11-05]. ISSN 0098-3063. Dostupné z: doi:10.1109/TCE.2019.2957987
- [67] LAPARRA, Valero, Johannes BALLÉ, Alexander BERARDINO a Eero P SIMONCELLI. Perceptual image quality assessment using a normalized Laplacian pyramid. *Electronic Imaging* [online]. Springfield (VA), 2016, 14 Feb 2016, 28(16), 1-6 [cit. 2022-12-04]. ISSN 2470-1173. Dostupné z: doi:10.2352/ISSN.2470-1173.2016.16.HVEI-103
- [68] *IRU-R BT.500-14: Methodologies for the subjective assessment of the quality of television images* [online]. Ženeva (CH): ITU, 2020, 14.10.2019, 1-102 [cit. 2022-11-06]. Dostupné z: <https://www.itu.int/rec/R-REC-BT.500-14-201910-I/en>
- [69] *IRU-T P.910: Subjective video quality assessment methods for multimedia applications* [online]. Ženeva (CH): ITU, 2022, 29.07.2022, 1-54 [cit. 2022-11-06]. Dostupné z: <https://www.itu.int/rec/T-REC-P.910-202207-I/en>
- [70] Cmake: cmake v3.25 x64. *Cmake* [online]. c2000-2022, 16 November 2022 [cit. 2022-11-30]. Dostupné z:

- https://github.com/Kitware/CMake/releases/download/v3.25.0/cmake-3.25.0-windows-x86_64.msi
- [71] Visual Studio 2022: Microsoft Visual Studio IDE v17.4.1 x64. *Microsoft* [online]. Redmond (WA), c2022, 15 Nov 2022 [cit. 2022-11-30]. Dostupné z: <https://visualstudio.microsoft.com/sk/downloads/>
- [72] KandaovR: Galéria. *KandaovR* [online]. Shenzhen (CN) [cit. 2022-12-05]. Dostupné z: <https://www.kandaovr.com/gallery/>
- [73] ESO: Image Archive: 360 Panorama. *ESO: European Southern Observatory* [online]. Mníchov (DE) [cit. 2022-12-05]. Dostupné z: <https://www.eso.org/public/czechrepublic/images/archive/category/360pano/>
- [74] Prehrávač MPC-HC: MPC-HC v1.9.24 x64. *VideoHelp* [online]. c1999-2022, 12 November 2024 [cit. 2022-12-09]. Dostupné z: <https://www.videohelp.com/download/MPC-HC.1.9.24.x64.exe>
- [75] Download FastStone Image Viewer: FastStone Image Viewer 7.7. *FastStone* [online]. c2022, 24 08 2022 [cit. 2022-12-21]. Dostupné z: <https://www.faststone.org/FSIVDownload.htm>
- [76] SITI: Command-line tool for calculating Spatial Information / Temporal Information according to ITU-T P.910. *GitHub* [online]. San Francisco (CA), c2023, 16 Aug 2019 [cit. 2023-05-07]. Dostupné z: <https://github.com/Telecommunication-Telemidia-Assessment/SITI>
- [77] FFmpeg v5.1.2 x64. *FFmpeg* [online]. 26 09 2022 [cit. 2022-11-30]. Dostupné z: <https://www.gyan.dev/ffmpeg/builds/ffmpeg-release-full.7z>
- [78] YUView: YUView v2.13. *GitHub* [online]. San Francisco (CA), c2022, 29 Apr 2022 [cit. 2022-12-05]. Dostupné z: <https://github.com/IENT/YUView/releases/download/v.2.13/YUView-Win-noautoupdate.zip>
- [79] Implementácia kodeku JPEG: libjpeg-turbo v2.1.4 x64. *Libjpeg-turbo* [online]. 24 October 2022n. 1., 12 08 2022 [cit. 2022-11-30]. Dostupné z: <https://sourceforge.net/projects/libjpeg-turbo/files/latest/download>
- [80] Implementácia kodeku HEIC/HEVC: HM v17.0. *Fraunhofer* [online]. Mníchov (DE), c2022, 19 10 2022 [cit. 2023-01-07]. Dostupné z: <https://vcgit.lhi.fraunhofer.de/jvet/HM/-/archive/HM-17.0/HM-HM-17.0.zip>
- [81] Implementácia kodeku AVIF: avifenc v0.11.1 aom v3.5.0. *GitHub* [online]. San Francisco (CA), 2008, 19 10 2022 [cit. 2022-11-30]. Dostupné z: <https://ci.appveyor.com/api/buildjobs/v39shvobkckxeo8y/artifacts/build%2Favifenc.exe>
- [82] Implementácia kodeku AVIF: avifdec v0.11.1 aom v3.5.0. *GitHub* [online]. San Francisco (CA), 2008, 19 10 2022 [cit. 2022-11-30]. Dostupné z: <https://ci.appveyor.com/api/buildjobs/v39shvobkckxeo8y/artifacts/build%2Favifdec.exe>
- [83] Implementácia kodeku JPEG XL: libjxl v0.7.0 x64. *GitHub* [online]. San Francisco (CA), 2008, 21 Sep 2022 [cit. 2022-11-30]. Dostupné z:

- <https://github.com/libjxl/libjxl/releases/download/v0.7.0/jxl-x64-windows-static.zip>
- [84] Implementácia kodeku VVC Intra/VVC: VTM v18.1. *Fraunhofer* [online]. Mníchov (DE), c2022, 24 10 2022 [cit. 2022-11-30]. Dostupné z: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/archive/VTM-18.1/VVCSoftware_VTM-VTM-18.1.zip
- [85] Implementácia kodeku VVC Intra/VVC: VVenC v1.7.0. *GitHub* [online]. San Francisco (CA), 2008, 30 Sep 2022 [cit. 2023-01-07]. Dostupné z: <https://github.com/fraunhoferhhi/vvenc/archive/refs/tags/v1.7.0.zip>
- [86] Implementácia kodeku VVC Intra/VVC: VVdeC v1.6.0. *GitHub* [online]. San Francisco (CA), 2008, 6 Sep 2022 [cit. 2022-11-30]. Dostupné z: <https://github.com/fraunhoferhhi/vvdec/archive/refs/tags/v1.6.0.zip>
- [87] VVenC vs VTM: Encoder Performance. *GitHub* [online]. San Francisco (CA), c2022, 12 Oct 2022 [cit. 2022-12-04]. Dostupné z: <https://github.com/fraunhoferhhi/vvenc/wiki/Encoder-Performance>
- [88] TensorFlow: Tensorflow-GPU Windows Native x64. *TensorFlow* [online]. Mountain View (CA), [2022], 6 September 2022 [cit. 2022-11-30]. Dostupné z: <https://www.tensorflow.org/install/pip>
- [89] PyTorch: PyTorch-GPU v1.13.0 Windows x64. *PyTorch* [online]. San Francisco (CA), c2022, 28 October 2022 [cit. 2022-12-19]. Dostupné z: <https://pytorch.org/get-started/locally/>
- [90] Python 3.10: Python v3.10.8 x64. *Python* [online]. Portland (OR), c2001-2022, 11 Oct 2022 [cit. 2022-11-30]. Dostupné z: <https://www.python.org/ftp/python/3.10.8/python-3.10.8-amd64.exe>
- [91] NVIDIA grafický ovládač: GeForce Game Ready Driver v526.98 x64. *NVIDIA* [online]. Santa Clara (CA), c2022, 16 11 2022 [cit. 2022-11-30]. Dostupné z: <https://www.nvidia.com/content/DriverDownloads/confirmation.php?url=/Windows/526.98/526.98-notebook-win10-win11-64bit-international-dch-whql.exe&lang=us&type=geforcem>
- [92] Anaconda: Anaconda Python v3.9 x64. *Anaconda* [online]. Austin (TX), c2022, 18 October 2022 [cit. 2022-11-30]. Dostupné z: https://repo.anaconda.com/archive/Anaconda3-2022.10-Windows-x86_64.exe
- [93] NVIDIA CUDA Toolkit: CUDA Toolkit v11.2 x64. *NVIDIA* [online]. Santa Clara (CA), c2022, 16 12 2020 [cit. 2022-11-30]. Dostupné z: https://developer.download.nvidia.com/compute/cuda/11.2.0/local_installers/cuda_11.2.0_460.89_win10.exe
- [94] NVIDIA CUDA Deep Neural Network: NVIDIA cuDNN v8.1.0 CUDA 11.2. *NVIDIA* [online]. Santa Clara (CA), c2022 [cit. 2022-11-30]. Dostupné z: https://developer.nvidia.com/compute/machine-learning/cudnn/secure/8.1.0.77/11.2_20210127/cudnn-11.2-windows-x64-v8.1.0.77.zip

- [95] NVIDIA TensorRT: NVIDIA TensorRT v7.2.3.4 CUDA 11.2 cuDNN 8.1.0. *NVIDIA* [online]. Santa Clara (CA), c2022 [cit. 2022-11-30]. Dostupné z: https://developer.nvidia.com/compute/machine-learning/tensorrt/secure/7.2.3/zip/TensorRT-7.2.3.4.Windows10.x86_64.cuda-11.1.cudnn8.1.zip
- [96] PyCUDA: pycuda v2019.1.1. *PyPI: The Python Package Index* [online]. Portland (OR), c2022, 10 Jul 2019 [cit. 2022-12-05]. Dostupné z: <https://pypi.org/project/pycuda/2019.1.1/>
- [97] Full Resolution Image Compression with Recurrent Neural Networks. *GitHub* [online]. San Francisco (CA), c2022, 16 Apr 2020 [cit. 2022-12-21]. Dostupné z: <https://github.com/tensorflow/models/tree/archive/research/compression>
- [98] Context-adaptive Entropy Model for End-to-end Optimized Image Compression. *GitHub* [online]. San Francisco (CA), c2022, 2 Apr 2020 [cit. 2022-12-21]. Dostupné z: https://github.com/JooyoungLeeETRI/CA_Entropy_Model
- [99] High-fidelity-generative-compression. *GitHub* [online]. San Francisco (CA), c2022, 5 Dec 2021 [cit. 2022-12-21]. Dostupné z: <https://github.com/Justin-Tan/high-fidelity-generative-compression>
- [100] LIC360. *GitHub* [online]. San Francisco (CA), c2022, 25 Oct 2022 [cit. 2022-12-21]. Dostupné z: <https://github.com/limuhit/360-Image-Compression>
- [101] BUDÁČ, Adam. Comparison of compression and decompression speeds for 360° images with resolutions up to 8K using conventional and deep learning algorithms. Proceedings I of the 29th Conference STUDENT EEICT 2023 [online]. [cit. 2023-05-19]. ISSN 2788-1334. Dostupné z: <https://www.eeict.cz/download> [online]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2023, 25 Apr 2023, 4 s. [cit. 2023-5-19]. ISSN 2788-1334. ISBN 978-80-214-6154-3. Dostupné z: <https://www.eeict.cz/download>. Prijaté k publikovaniu
- [102] KUFA, Jan a Adam BUDAC. Quality comparison of 360° 8K images compressed by conventional and deep learning algorithms. 2023 33rd International Conference Radioelektronika (RADIOELEKTRONIKA) [online]. Ženeva (CH): IEEE, 2023, 19 April 2023, 1-4 [cit. 2023-05-06]. ISBN 979-8-3503-9834-2. Dostupné z: doi:10.1109/RADIOELEKTRONIKA57919.2023.10109066
- [103] VQMT: Video Quality Measurement Tool. *EPFL* [online]. Lausanne (CH), c2022, 30 3 2013 [cit. 2022-12-03]. Dostupné z: <https://www.epfl.ch/labs/mmsp/dnloads/vqmt/>
- [104] 360tools: Konverzia projekcie a metriky 360° obrazu. *GitHub* [online]. San Francisco (CA), c2022, 4 Oct 2014 [cit. 2022-12-03]. Dostupné z: <https://github.com/Samsung/360tools>
- [105] MATLAB: MATLAB R2022b x64. *MathWorks* [online]. Natick (MA), c1994-2022, 28 Sep 2021 [cit. 2022-12-19]. Dostupné z: <https://www.mathworks.com/products/matlab.html>

- [106] FSIMc: Implementácia metriky. *The Hong Kong Polytechnic University: Faculty of Engineering: Department of Computing* [online]. Hong Kong (HKG), c2022, 20 Jan 2011 [cit. 2022-12-03]. Dostupné z:
<https://web.comp.polyu.edu.hk/cslzhang/IQA/FSIM/Files/FeatureSIM.m>
- [107] GMSD: Implementácia metriky. *The Hong Kong Polytechnic University: Faculty of Engineering: Department of Computing* [online]. Hong Kong (HKG), c2022, 2014 [cit. 2022-12-03]. Dostupné z:
<http://www4.comp.polyu.edu.hk/~cslzhang/IQA/GMSD/GMSD.m>
- [108] Netflix VMAF: vmaf v2.3.1. *GitHub* [online]. San Francisco (CA), 2008, 11 04 2022 [cit. 2022-12-03]. Dostupné z:
<https://github.com/Netflix/vmaf/releases/download/v2.3.1/vmaf.exe>
- [109] Oculus Rift Setup: Get Started with Rift. Oculus Rift Setup [online]. Menlo Park (CA), c2023, 2023 [cit. 2023-05-05]. Dostupné z:
<https://www.oculus.com/rift/setup/>
- [110] BJØNTEGAARD, Gisle. Calculation of average PSNR differences between RD-curves [online]. Ženeva (CH): ITU, 2001, 4 April 2001, 4 [cit. 2023-05-07]. Dostupné z: https://www.itu.int/wftp3/av-arch/video-site/0104_Aus/VCEG-M33.doc
- [111] Bjøntegaard-Delta Interpolation: BD-rate implementations in Python. *GitHub* [online]. San Francisco (CA), c2023, 24 Feb 2022 [cit. 2023-05-07]. Dostupné z: <https://github.com/FAU-LMS/bjontegaard>

ZOZNAM SYMBOLOV A SKRATIEK

Skratky:

ACR	Absolute category rating
ACR-HR	Absolute category rating with hidden reference
ADST	Asymmetric discrete sine transform
AE	Autoencoder
AI	Artificial intelligence
AN-SNR	Anti-Noise Signal to Noise Ratio
ANN	Artificial neural network
ANS	Asymmetric numeral systems
AOMedia	Alliance for Open Media
AR	Augmented reality
AV1	AOMedia Video 1
AVIF	AV1 image file format
AVX	Advanced vector extensions
BD	Bjøntegaard-Delta
BPP	Bits per pixel
CABAC	Context-adaptive binary arithmetic coding
CAEMfE2EOIC	Context-adaptive Entropy Model for End-to-end Optimized Image Compression
CAVLC	Context-adaptive variable-length coding
CB	Coding block
CCITT	Consultative Committee on International Telegraph and Telephone
CDEF	Constrained directional enhancement filter
CI	Confidence interval
CMP	Cubemap projection
CNN	Convolutional neural network
CPP	Craster parabolic projection
CPP-PSNR	Craster parabolic projection Peak Signal to Noise Ratio
CPU	Central processing unit
CU	Coding unit
CTB	Coding tree block
CTU	Coding tree unit
DCT	Discrete cosine transform
DL	Deep learning
DLM	Detail loss metric
DNN	Deep neural network
DPCM	Differential pulse code modulation

DSIS	Double stimulus impairment scale
DST	Discrete sine transform
DWT	Discrete wavelet transform
EAP	Equal-area projection
ELU	Exponential linear unit
ERP	Equirectangular projection
FF	Feed-forward
FPS	Frames per second
FR	Full reference
FRICwRNN	Full Resolution Image Compression with Recurrent Neural Networks
FSIM	Feature similarity index
FSIMc	Feature similarity chrominance
GAN	Generative adversarial network
GMS	Gradient magnitude similarity
GMSD	Gradient magnitude similarity deviation
GPU	Graphics processing unit
GRU	Gated recurrent unit
GUI	Graphical user interface
HDR	High dynamic range
HEIC	High efficiency image coding
HEVC	High efficiency video coding
HiFiC	High-Fidelity Generative Image Compression
HVS	Human visual system
HW	Hardware
IDCT	Inverse discrete cosine transform
ISO	International Organization for Standardization
JPEG	Joint Photographic Experts Group
LIC360	End-to-End Optimized 360° Image Compression
LPC	Linear predictive coding
LPIPS	Learned perceptual image patch similarity
LZ	Lempel-Ziv
MCPD	Mean co-located pixel difference
MCU	Minimum coded unit
ML	Machine learning
MLP	Multi-layer perceptron
MPEG	Moving Picture Experts Group
MS-SSIM	Multiscale structural similarity
MSBAC	Multi-symbol binary arithmetic coding
MSE	Mean square error

NaN	Not a number
NLPD	Normalized Laplacian pyramid distance
NN	Neural network
NR	No reference
OS	Operating system
PC	Personal computer
PCM	Pulse code modulation
PLCC	Pearson's linear correlation coefficient
PSNR	Peak Signal to Noise Ratio
PSNR-HVS	Peak Signal to Noise Ratio – Human Visual System
PSNR-HVS-M	Peak Signal to Noise Ratio – Human Visual System – Masking model
PSU	Power supply unit
PU	Prediction unit
RAM	Random access memory
ReLU	Rectified linear unit
RD	Rate-distortion
RLE	Run-length encoding
RNN	Recurrent neural network
RR	Reduced reference
S-PSNR	Spherical Peak Signal to Noise Ratio
SI	Spatial index
SIMD	Single instruction multiple data
SDR	Standard dynamic range
SNR	Signal to Noise Ratio
SROCC	Spearman's rank-order correlation coefficient
SSD	Solid state disk
SSIM	Structural similarity
SVM	Support vector machine
SW	Software
TI	Temporal index
TSP	Truncated sphere pyramid projection
TU	Transform unit
VarDCT	Variable-blocksize discrete cosine transform
VIFp	Visual information fidelity pixel domain
VMAF	Video multi-method assessment fusion
VMSE	Viewport-based mean square error
VR	Virtual reality
VRAM	Video random access memory
VSNR	Visual Signal to Noise Ratio

VSSIM	Viewport-based structural similarity
WCG	Wide color gamut
WS-PSNR	Weighted to Spherically uniform Peak Signal to Noise Ratio

ZOZNAM PRÍLOH

PRÍLOHA A -	DATASET OBRÁZKOV	100
PRÍLOHA B -	KOMPRESNÉ ARTEFAKTY SPÔSOBENÉ KODEKMI	101

Príloha A - Dataset obrázkov



in01, SI = 49,8



in02, SI = 61,68



in03, SI = 83,94



in04, SI = 50,9



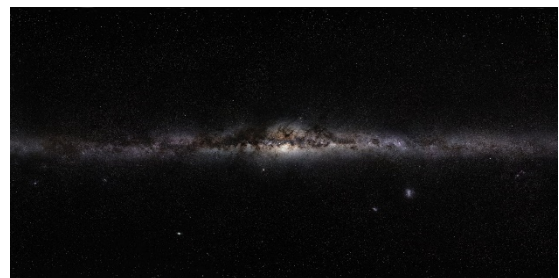
in05, SI = 68,61



in06, SI = 72,92



in07, SI = 45,96



in08, SI = 98,95



in09, SI = 50,74



in10, SI = 98,15

Príloha B - Kompresné artefakty spôsobené kodekmi



Originál



JPEG, kvalita 9



JPEG XL, kvalita 5



HEIC, kvalita 49



AVIF, kvalita 60



VVC Intra, kvalita 46



FRICwRNN, kvalita 1



CAEMfE2EOIC, kvalita 1



HiFiC, kvalita 1



LIC360, kvalita 1