FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání diplomové práce

Název:	Automatizovaný střih videa zápasů beachvolejbalu
Student:	Bc. Justýna Frommová
Vedoucí:	Ing. Jakub Novák
Studijní program:	Informatika
Obor / specializace:	Znalostní inženýrství
Katedra:	Katedra aplikované matematiky
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

Cílem práce je navrhnout algoritmus pro automatický střih videa beachvolejbalového zápasu na jednotlivé výměny. Aktuální zpracování beachvolejbalových zápasů pracuje právě s výměnami za účelem zmenšení objemu dat. Vše je prováděno manuálně. Kvantita odehraných zápasů na světové úrovni je v řádu tisíců, zautomatizování střihu videa velmi urychlí další práci pro zisk statistik.

1) Seznamte se s problematikou beachvolejbalu, jeho statistik a možnými softwarovými nástroji pro zpracování sportovních videí.
2) Proveďte rešerši v oblasti algoritmů vhodných pro automatický střih (např. pomocí metod klasifikace videí).
3) Najděte a prozkoumejte možné datasety zápasů, zhodnoťte použitelnost.
4) Navrhněte a implementujte několik algoritmů (např. 3) pro střih videa s použitím vybraných datasetů.
5) Otestujte, vizualizujte a vyhodnoťte výsledky algoritmu.
6) Diskutujte výsledky.

Elektronicky schválil/a Ing. Magda Friedjungová, Ph.D. dne 31. ledna 2024 v Praze.

Diplomová práce

AUTOMATIZOVANÝ STŘIH VIDEA ZÁPASŮ BEACHVOLEJBALU

Bc. Justýna Frommová

Fakulta informačních technologií
Katedra aplikované matematiky
Vedoucí: Ing. Jakub Novák
9. května 2024

Obsah

Seznam obrázků

Seznam tabulek

Chtěla bych poděkovat především vedoucímu Ing. Jakubu Novákovi za ochotu, cenné rady a mentoring, nejen po dobu psaní práce, ale celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 9. května 2024

Abstrakt

Práce se zabývá automatizovaným střihem videa zápasů beachvolejbalu. Byl vytvořen dataset 324 videí zápasů z beachvolejbalové sezóny 2023 a 88 zápasů ze sezóny 2024, což odpovídá více než 300 hodinám záznamu. V rámci vlastní realizace byly implementovány tři algoritmy automatizovaného střihu videa pomocí optical flow, 2D klasifikace a 2.5D klasifikace. Model nejlépe vyhodnoceného přístupu – 2.5D klasifikace – je postaven na architektuře EfficientNetV2-B3 a byl trénován na 221 záznamech. Střih dosahuje 93% přesnosti na jednotlivých snímcích videa, IoU skóre intervalů střihu odpovídá 77,2 %. Precision na úrovni jednotlivých výměn, tedy pravděpodobnost, že predikovaná výměna je skutečnou výměnou, dosahuje 99,6 % a model sestříhá 64 % zápasů bez chyby v počtu výměn. Průměrná chyba začátku či konce výměny nepřesahuje 2 sekundy.

Klíčová slova klasifikace videa, detekce výměn, střih videa, beachvolejbal, EfficientNetV2, optical flow

Abstract

The thesis deals with the automated cutting of beach volleyball matches. A dataset comprising 324 match videos from the 2023 beach volleyball season and 88 matches from the 2024 season was created, totalling over 300 hours of footage. Three algorithms for automated video cutting were implemented: optical flow, 2D classification and 2.5D classification. The model employing the most successful approach – 2.5D classification – is built upon the EfficientNetV2-B3 architecture and was trained on 221 recordings. The classification accuracy on individual video frames achieves a 93% accuracy, with an IoU score for rally intervals of 77.2 %. Precision at the level of individual rallies, indicating the probability that a predicted rally corresponds to an actual rally, reaches 99.6 %, and the model successfully edits 64 % of matches without errors in the total number of rallies. The average error in start or end of a rally does not exceed 2 seconds.

Keywords video classification, rally detection, video cutting, beach volleyball, EfficientNetV2, optical flow

Seznam zkratek

BN	Batch normalizace
CNN	Konvoluční neuronová síť
FC	Fully-connected vrstva
FDR	False discovery rate
FIVB	Fédération Internatonal de Volleyball
FPS	Frames per second
FN	False negative
FP	False positive
NAS	Neural architecture search
SSIM	Structural similarity index measure
TP	True positive

Úvod

Důležitou součástí úspěchu vrcholového beachvolejbalového týmu není pouze fyzická příprava, ale i taktické rozhodnutí. Každý tým má ve svém realizačním týmu skupinu statistiků, kteří se starají o převádění jednotlivých zápasů na data, jež slouží jako základ pro tvorbu strategie do nadcházejících zápasů nebo pro analýzu vlastních výkonů. Během sezóny tým beachvolejbalistů odehraje kolem 10 turnajů, kde minimální počet zápasů se pohybuje kolem 2 zápasů, v případě kvalifikace je minimální počet snížen o jeden zápas. Práce statistika spočívá v čistě ruční analýze zápasů naživo, nebo ze záznamů. Pro propojení statistických informací zápasu s vizuálním podkladem, je běžnou praxí rozstříhat záznam zápasu na jednotlivé výměny.

Práce se zabývá automatizací procesu střihu videí beachvolejbalových zápasů na jednotlivé výměny. Počet celkově odehraných zápasů nejvyšší kategorie během jedné sezóny se pohybuje v řádu tisíců jednotek. Navíc v průběhu vyřazovací části turnaje, tzv. play-off, je nutné provést analýzu a střih zápasu v rámci několika hodin. Metody navržené v práci algoritmicky řeší střih videa za účelem zefektivnění a omezení manuální práce s videem.

Cílem práce je využití počítačového vidění, umělé inteligence a zpracování obrazu k vytvoření několika algoritmů pro automatizaci střihu videa beachvolejbalového zápasu, jež jsou vzájemně porovnány a vyhodnoceny. Vyhodnocování probíhá na vytvořených datasetech z videí zápasů a jejich již nastříhaných částí. Nejlépe hodnocený algoritmus a výsledky práce poslouží jako podklad pro integraci automatizace střihu videa do stávajícího produktu Beach-Data.

Kapitola 1
Rešerše

Každodenně narůstá objem záznamů sportovních událostí, které obsahují části, které nejsou pro diváky s pouze sportovními zájmy atraktivní. Z tohoto důvodu jsou vyvíjeny aplikace, jež automaticky zkracují záznamy sportovních akcí a selektují ty části, které mají pro konkrétní sportovní oblast či zápas největší relevanci. Pro efektivní zpracování sportovních videí se stále více uplatňují softwarové nástroje využívající technologie umělé inteligence.

1.1 Softwarové nástroje pro zpracovaní sportovních videí

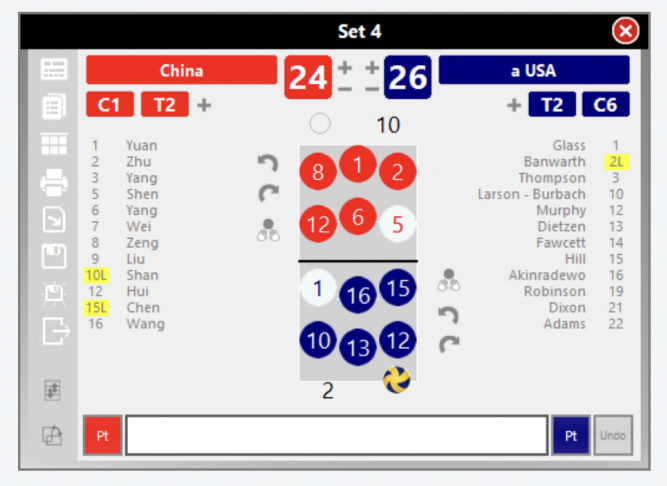
V kontextu současného vývoje technologií v oblasti sportovního tréninku a analýzy se zpracování velkého množství sportovních videí stává klíčovým prvkem pro optimalizaci výkonu profesionálních sportovců.

1.1.1 Data Project

Nejpoužívanějším softwarem pro analýzu zápasů volejbalu je software **Data Volley** firmy Data Project [1]. Data Volley je využíván statistiky nejen profesionálních týmů, ale i týmů národních napříč celým světem [2, 3, 4, 5]. Software umožňuje skautování real-time během zápasu s vytvářením statistických výstupů pro trenéry. Software lze nastavit i pro analyzování beachvolejbalu upravením počtu hráčů a velikostí hřiště. Verze Data Volley 4 umožňuje integraci videa zápasu bez potřeby dalších programů. Integrace umožňuje ručně namapovat jednotlivé statistiky na záznam zápasu.

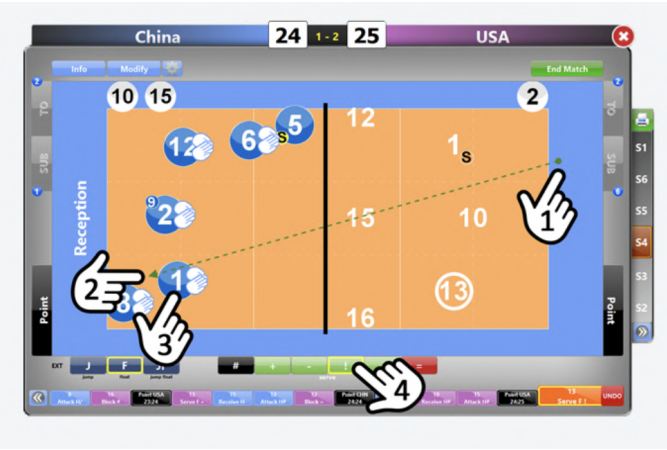
Skautování je prováděno pomocí kódů. Kód akce "*Hráč hostujícího týmu č.1 podává skákaným servisem ze zóny 1 do zóny 6, kde hráč hostujících č. 8 perfektně míč přihrává.*"je `*01SQ-16;*08RQ#`. Základním kódem je prvních 6 znaků, informace o zónách patří do kódu pokročilého. Prostředí a ovládání software klade větší důraz na efektivitu než na uživatelskou přívětivost (viz. Obrázek 1.1)

Jednou z klíčových funkcí programu Data Volley je generování reportů z naskautovaných dat. V PRO verzi programu je možné reporty personalizovat podle potřeb uživatele. Report obsahuje nejen statistické výstupy, ale také zónovou analýzu, která zahrnuje distribuci útoků, servisů a příjmů v rámci různých zón hřiště, identifikaci nejčastějších směrů útoků z konkrétní zóny a další relevantní informace.



Obrázek 1.1 Uživatelské rozhraní software Data Volley [1].

Zjednodušenou verzí od stejné firmy je software **Click&Scout** [6]. Aplikace upřednostňuje jednoduchost skautování, které není prováděno pomocí kódů, ale klikáním přímo na virtuální hřiště (viz. Obrázek 1.2). Výstup je zpětně kompatibilní se softwarem Data Volley, jelikož jednotlivé kliky jsou reprezentovány stejnými kódy, které Data Volley využívá.



Obrázek 1.2 Uživatelské rozhraní software Click&Scout [6].

1.1.2 Beach Data

Konkurentem v oblasti beachvolejbalové analýzy je český software **Beach Data** [7]. Beach Data se na rozdíl od Data Volley zaměřuje pouze na beachvolejbal. Aplikace je ovládána stylem ”drag&drop”na virtuálním hřišti a cílí na uživatelskou přívětivost. Aplikace je momentálně dostupná pouze pro zařízení Apple iPad.

Největším rozdílem a výhodou oproti ostatním aplikacím je sdílená databáze zápasů FIVB. Beach Data jsou synchronizována s FIVB databází zápasů, včetně údajů o hráčích. Skautování zápasů probíhá ručně po skončení zápasu z video záznamu, přičemž zkušenější skauti jsou schopni zápas zaznamenávat v reálném čase. Po dokončení skautování zápasu jsou data nahrána na ser-

ver a skaut zajišťuje synchronizaci záznamu zápasu s naskautovanými daty pomocí sestříhání záznamu v nadstavbové aplikaci Beach Data Video. Zákazník používající aplikaci nemá přístup pouze k zápasům, které si sám oskautoval, ale také ke všem již naskautovaným videím ze světové série či národní tour.

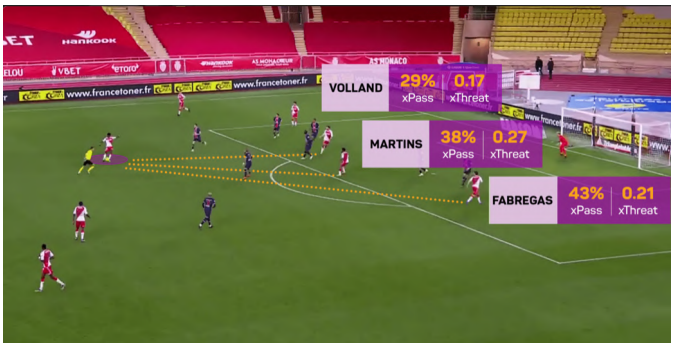
Během skautování jsou získávána data o pozicích hráčů při odbití míče, typ zpracování míče, nebo výměna začíná po oddechovém čase. Beach Data kladou důraz na správnost a přesnost dat, proto skautování provádějí lidé, kteří beachvolejbalu rozumí. Znalost sportu je žádoucí i pro samotné skautování, jelikož se zde hodnotí např. i kvalita nahrávky, typ útoku apod.

1.1.3 Stats Perform

Stats Perform [8] je firmou, jejichž platforma je využívána celosvětově špičkovými sportovními týmy, televizemi i sázkovými kanceláři. Stats Perform se zaměřuje na livestreamy, data pro sázkové kanceláře, a především na analýzu sportovních dat. Po dobu již 40 let sbírají sportovní data o fotbalu, basketbalu, kriketu a mnoho dalších. Data byla a stále jsou zpracovávána lidskou sílou. Od roku 2019 využívá Stats Perform umělou inteligenci pro vyhodnocování dat, vytváření analýz, predikci výsledků zápasů, pro lepší porozumění komplexity daného sportu a hledání vzorů pro zlepšení výkonu. Konkrétně se jedná například o pravděpodobnost, že daná přihrávka ve fotbalu bude proměněna v gól. Pravděpodobnost, že padne gól na základě podobnosti situace nebo predikce vývoje skóre v rugby v závislosti na počtu odehraných minut.

Dalším konkrétním využití umělé inteligence je tracking hráčů a míče po hřišti. Tyto data jsou využívána pro televizní grafiku, měření vzdáleností nebo při jestřábím oku, ať už se jedná o tenis, volejbal nebo basketbal.

V roce 2022 představil nový koncept systému **Opta Vision** zaměřující se na fotbal. Data jednotlivých akcí zápasu jsou získávána ručně a obecně jim chybí kontext, proč se daná akce odehrává. Naopak data získávaná z trackingu nesou informaci o poloze každého hráče na hřišti a míče, ale neobsahuje informaci o prováděné akci. Koncept Opta Vision spojuje oba typy dat dohromady, čímž umožňuje tvořit analýzy polohového charakteru s větším kontextem. Přibyla tím například analýzu situací, kdy přihrávka úspěšně protne obrannou formaci soupeře, nebo real-time predikce přihrávky a její úspěšnosti (viz. Obrázek 1.3). Nový koncept není plně autonomní a analytičtí odborníci jsou stále součástí procesu tvorby a kontroly dat.



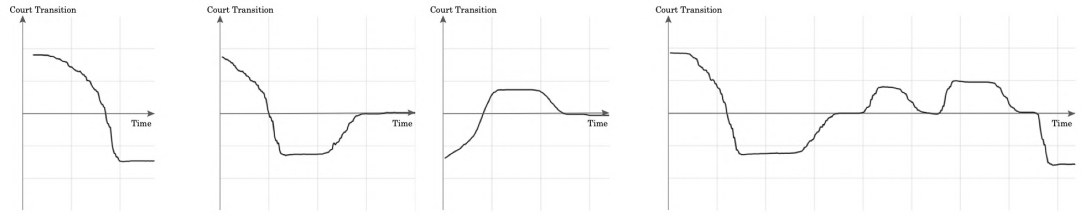
Obrázek 1.3 Ukázka real-time predikce [8].

1.2 Automatický střih videa

Existují dva přístupy k řešení střihu sportovních videí. První přístup spočívá v (ručním) vystřihování částí z celého záznamu sportovních událostí, které jsou charakteristické a zajímavé pro daný sport. Druhý přístup pak zahrnuje pohled na automatický střih videa jako na problém supervizované klasifikace.

1.2.1 Střih jako sumarizace

Výzkumný tým z Waseda univerzity v Japonsku se v roce 2016 a 2017 věnoval sumarizaci sportovních videí raketových sportů (tenis, badminton, stolní tenis) a volejbalu [9, 10]. Obě práce využívají sestříhaná videa vysílaná do komerční sféry obsahující záběry výměn, zpomalené opakování akcí a záběry na fanoušky. Mezi danými záběry je zřetelný střih, který slouží k segmentaci videa na n částí. Shluky vysegmentovaných částí jsou vytvářeny dle odlišnosti HSV histogramu. Rozpoznání částí odpovídající záběru výměny je na základě detekce bílých čar ohraničujících hřiště. V případě volejbalu lze dle pozice kamery rozpoznat, na jaké straně hřiště je zrovna míč, jelikož autoři předpokládají, že kamera zabírá vždy tu polovinu hřiště, kde se míč nachází. Změny polohy míče určují o jaký typ výměny se jedná – pouze servis, servis a útok, delší výměna (viz. Obrázek 1.4). Výsledné záběry výměn mají přidělenou váhu dle délky, aby výsledná sumarizace obsahovala zajímavější výměny. Precision algoritmu [10] je 0,992 a recall 0,862.



Obrázek 1.4 Ukázka typů přechodu míče – vlevo pouze servis, vpravo dlouhá výměna, prostřední dva snímky znázorňují standardní servis a útok [9].

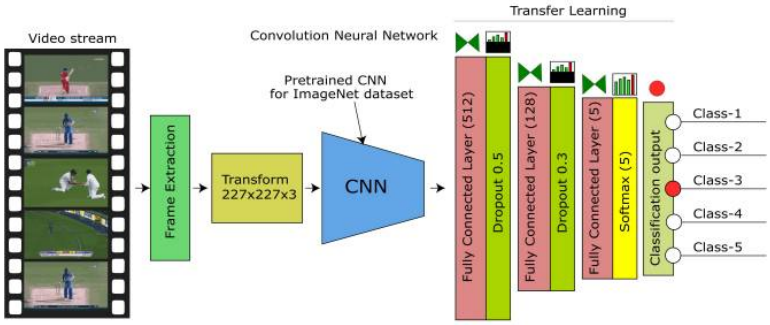
Zhao a kolektiv v roce 2012 využili pro sumarizaci broadcastové záběry s přechodovými obrazovky mezi střihy slow-motion záběrů. Přechodové obrazovky uvádějící a zakončující slow-motion záběry obsahují vždy logo dané sportovní akce, které je v obraze detekované [11].

Peker a kolektiv v roce 2001 předpokládali, že nejzajímavější momenty sportovních zápasů je možné generovat pomocí kvantity časového vzorce pohybu, který koreluje s charakteristikou daného sportu [12]. Pro vyhodnocení míry akce v n-tém okamžiku videa jsou využity MPEG-7 pohybové deskriptory [13]. Křivka míry akce je vyhlazena aplikováním klouzavého průměru a mediánového filtru s velikostí kernelu 17 pro golf, 25 pro fotbal a prahována adaptivním prahováním. Nejzajímavějšími momenty jsou intervaly s vysokou aktivitou, které předchází aktivita malá. Algoritmus kompenzuje nižší preciznost rychlostí výpočtu.

1.2.2 Hluboké učení – klasifikace

Následující sekce obsahuje možné způsoby supervizované klasifikace videa, které lze převést na problém binární klasifikace, zda část videa – snímek, shluk snímků – patří do střihu či ne.

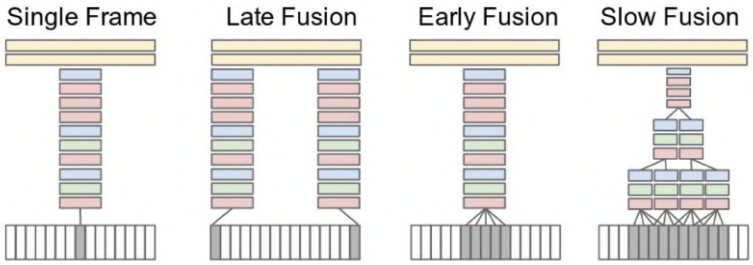
Rafiq a kolektiv v roce 2020 využili transfer learningu ke klasifikaci scén za účelem sumarizace sportovního videa, konkrétně zápasů indického kriketu [14]. Byl využit model AlexNet předtrénovaný na datasetu ImageNet-1k, k němuž byla přidána klasifikační hlava s třemi fully-connected vrstvami a dvěma dropout vrstvami (viz. Obrázek 1.5). Vstupní video, které má snímkovací frekvenci 30 snímků za sekundu, je převedeno na snímky se snímkovací frekvenci 6 snímků za sekundu a do modelu vstupují jako jednotlivé obrázky v rozlišení 227×227 pixelů ve formátu RGB. Video je převedeno na seznam snímků a klasifikace probíhá na jednotlivých snímcích, čímž je zanedbána časová závislost mezi snímky. Výsledná klasifikace scény je ohodnocena pomocí shlukování stejně vypredikované hodnoty do prvního výskytu signifikantní změny v ohodnocení. Model klasifikuje pět různých scén (odpalování, bowling, hranice, fanoušci, detailní záběr) s přesností 99,26 % a precision 99,27 %.



Obrázek 1.5 Ukázka architektury klasifikace [14].

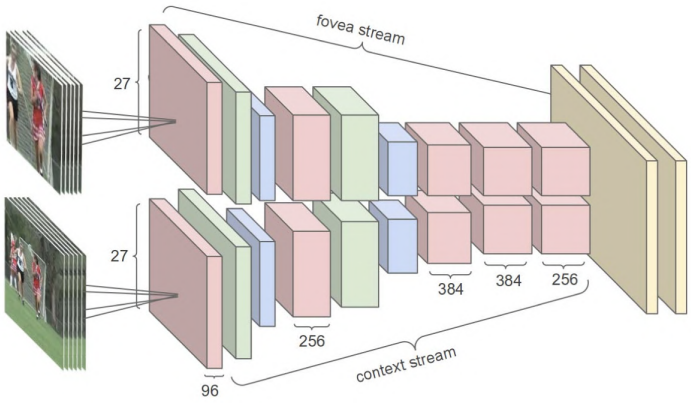
Karpathy a kolektiv v roce 2014 [15] evaluovali využití konvolučních sítí na velkém množství různorodých videí, konkrétně na miliónovém datasetu Youtube videí obsahující 487 tříd (Sports-1M dataset). Autoři video považují za množinu fixně velkých klipů a prověřují 3 typy zakomponování časové informace do konvolučních modelů, zobrazeno také na Obrázku 1.6.

- **Early fusion** – modifikací prvního filtru konvoluční vrstvy na $11 \times 11 \times 3 \times T$, kde T je počet snímků určující časovou závislost, se informace o časové závislosti dostane do modelu v první vrstvě a poté je časová informace ztracena.
- **Late fusion** – zpracování odděleně dvou snímků konvoluční sítí, která sdílí váhy a následné spojení pomocí fully-connected vrstvy.
- **Slow fusion** – kombinace předchozích typů, postupná fúze časové informace.



Obrázek 1.6 Typy fúzí časové informace skrz model [15].

Trénování milionového datasetu videí i na rychlém GPU trvá dlouho, proto autoři navrhli metodu, jak urychlit čas tréninku, ale zároveň zachovat kvalitu. Byla navržena architektura, která má 2 vstupy: snímek zmenšený na poloviční rozlišení a výstřižek o velikosti poloviního rozlišení středového regionu snímku (viz. Obrázek 1.7). Nejlepších výsledků v experimentech dosahoval typ fúze "Slow". "Single-frame"neboli klasifikace jednotlivých snímků byla též konkurenceschopná.



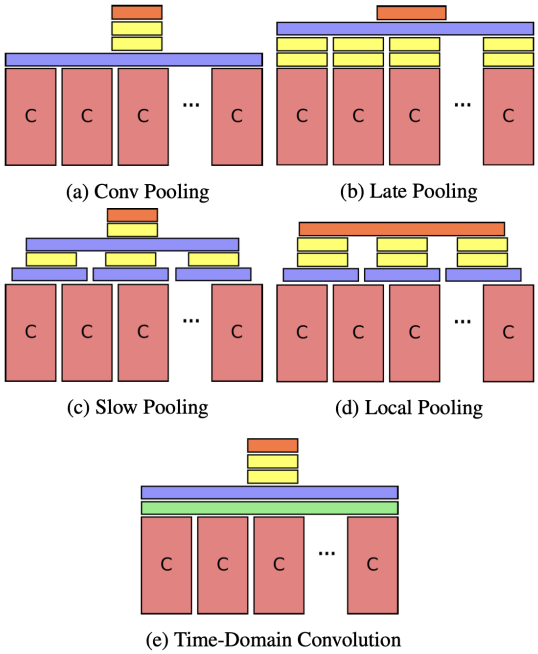
Obrázek 1.7 Multiresolution model CNN [15].

K. Simonyan a A. Zisserman v roce 2014 navrhli dvouvětvou CNN architekturu, kde jedna větev je určená na zisk prostorové a obrazové informace a druhá extrahuje časově závislé informace [16]. Časová složka je získávána metodou optical flow a je propagována modelem pomocí zvětšení hloubky konvolučních filtrů. Výsledná predikce je fúzí obou větví průměrováním nebo výstupem natrénovaného multi-class SVM modelu.

Ng a kolektiv v roce 2015 navrhli dva přístupy klasifikace až dvouminutových videí s využitím konvolučních neuronových sítí (CNN), feature poolingu nebo LSTM [17]. Feature pooling model zpracovává nezávisle jednotlivé snímky pomocí CNN a následně je zkombinuje dohromady pomocí jednoho z pěti typů poolingu. Na obrázku 1.8 jsou zobrazeny jednotlivé typy poolingu. Conv pooling provádí max-pooling přes všechny snímky videa, naopak late pooling napřed výstup předzpracuje FC vrstvami a poté je aplikován max-pooling. Pro extrakci příznaků byly v experimentu porovnány modely AlexNet a GoogleNet. Vstupní video je vzorkováno s frekvencí jeden snímek za sekundu. Jelikož s nízkou snímkovací frekvencí dochází ke ztrátě implicitní pohybové informace, je tento nedostatek kompenzován optical flow snímky sousedících snímků, které jsou druhým vstupem modelu (viz. Obrázek 1.9).

Váhy natrénovaného klasifikačního modelu obsahující feature pooling vrstvu s nižším počtem vstupních snímků byly použity pro trénink modelu s vyšším počtem vstupních snímků, konkrétně 1-30-120 snímků. Optical flow snímky jsou vytvářeny se vzorkovací frekvencí 15 snímků za sekundu. Nejlepších výsledků na datasetu Sports-1M bez vstupu optical flow dosahoval CNN model GoogleNet s aplikací Conv poolingu. Po přidání optical flow snímků do pipeline nejlepších výsledků na stejném datasetu dosahuje GoogleNet v kombinaci s LSTM.

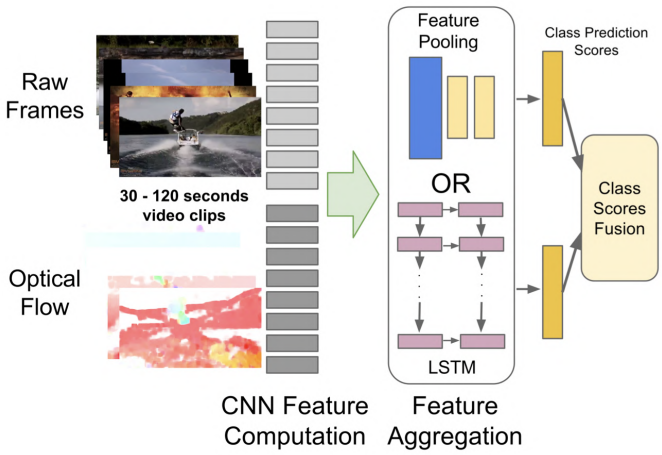
Tran a kolektiv v roce 2015 navrhli využití 3D konvoluce k naučení obrazovo-časových příznaků [18]. Model C3D obsahuje osm 3D konvolučních vrstev s kernelem o velikosti $3 \times 3 \times 3$, pět max-pooling vrstev a dvě FC vrstvy se softmax aktivací a je natrénovaný na Sports-1M datasetu.



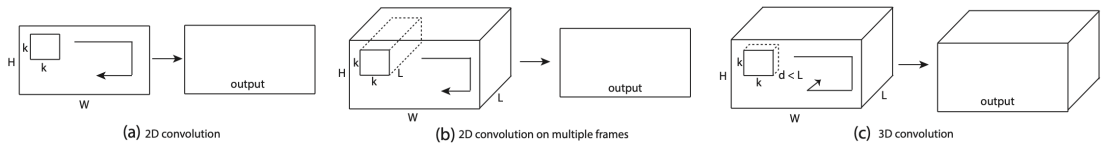
Obrázek 1.8 Typy feature poolingů [17]: Konvoluční vrstvy jsou vyznačeny písmenem C. Modrý, zelený, žlutý a oranžový obdélník znázorňuje max-pooling, time-domain konvoluci, FC vrstvu a softmax.

Huang a kolektiv v roce 2021 analyzovali vhodnost správného momentu na střih v nezeditovaném videu [19]. Výstup modelu je složen ze tří vstupů – množina 16 RGB snímků, labely z Mask R-CNN a optical flow. Je využita 3D Resnet architektura a natrénovány dva modely – klasifikační pro ohodnocení, zda daný klip je vhodný pro začátek/konec střihu. Jelikož okamžik správného okamžiku pro střih videa může trvat několik snímků, je aplikována metoda label smoothingu. Autoři pro trénink navrhli augmentaci časové složky videa využívající náhodného vzorkování z původní snímkovací frekvence na frekvenci menší. První a poslední snímek je daný, ale prostředních 14 je náhodně vybráno.

Porovnání mezi 2D a 3D konvolucí je viditelné na následujícím obrázku (Obrázek 1.10). 2D konvoluce na více konkatenovaných snímcích je dále v práci nazývána **2.5D konvolucí**.



Obrázek 1.9 Shrnutí metody klasifikace videa [17].



Obrázek 1.10 Porovnání typů konvoluce [18].

Kapitola 2
Teorie

V kapitole jsou teoreticky rozebrány metody a algoritmy použité v analytické a praktické části práce včetně vysvětlení pravidel a statistik beachvolejbalu.

2.1 Beachvolejbal

▶ **Definice 2.1** (Beachvolejbal)**.** *Beachvolejbal je míčový sport hraný dvěma týmy na píseč-ném kurtu přepůleném sítí. Tým má 3 doteky na vrácení míče k soupeři (včetně doteku bloku) [20].*

Turnaje na světové úrovni jsou pořádány světovou volejbalovou federací FIVB[1]. Federace má mimo jiné na starosti zveřejňování aktuálního světového žebříčku, definici olympijské kvalifikace, podmínky pořádání turnajů či pravidla samotné hry.

2.1.1 Pravidla

Beachvolejbalový zápas se skládá z jednotlivých výměn. Výměna je sekvence akcí vykonávaných hráči od podání až po ukončení zapísknutím rozhodčího. Dokončená akce vede k zisku bodu jednoho z týmu. Pokud bod získá tým začínající na servisu, servis zůstává u stejného týmu. Naopak pokud bod získá tým začínající na příjmu, při další výměně začíná na servisu.

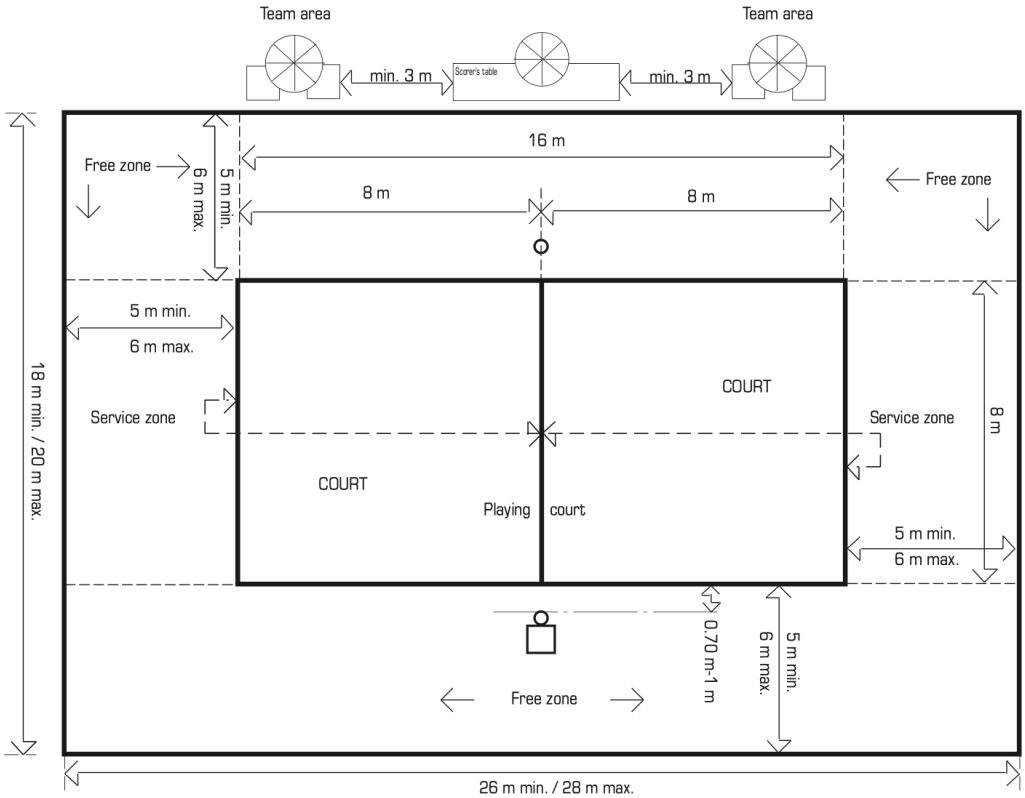
Zápas lze dále dělit na jednotlivé sety. Set vyhraje ten tým, který dříve docílí 21 bodů s minimálním rozdílem 2 bodů. To znamená, že v případě stavu 20:20 je zaručené trvání setu o další dvě výměny a možný konečný výsledek by mohl být 22:20. Zápas vyhraje ten tým, který vyhraje dva sety. V případě remízy 1-1 na sety následuje rozhodující třetí set, který se hraje pouze do 15 bodů s minimálním rozdílem 2 bodů.

V průběhu zápasu mají oba týmy možnost přerušit hru a vzít si oddechový čas jednou za set. Doba oddechového času je 30 vteřin. V prvním a druhém setu je navíc automaticky přidán technický oddechový čas při součtu bodů obou týmu rovnému 21 bodů. Trvání technického oddechového času je také 30 vteřin. Při oddechovém času hráči musí opustit kurt a jít do hráčské zóny, typicky na okraji kurtu. Po skončení setu je pauza 1 minuta.

[1] https://www.fivb.com/

Jelikož se plážový volejbal převážně hraje na venkovních kurtech, kde přírodní podmínky mohou velmi ovlivnit hru, byl zaveden systém střídání stran. Změna stran týmů je prováděna každých 7 bodů, v případě třetího setu každých 5 bodů.

Velikost kurtu a jednotlivé vzdálenosti mezi zónami jsou definované na obr 2.1. Hrací kurty musí vyhovovat definicím na všech oficiálních turnajích. Regulovaný je i vzhled sítě. Výška sítě je 2,43 metrů pro mužskou kategorii, 2,24 metrů pro kategorii ženskou. Kromě výšky je i definována délka, výška a velikosti ok sítě [20].



Obrázek 2.1 Diagram hrací zóny [20].

2.1.2 Statistiky

Týmy pohybující se v elitní špičce světového žebříčku jsou fyzicky a výkonnostně na podobné úrovni. Finální výsledek zápasu je proto i velmi ovlivňován taktickou přípravou týmu. Každý tým má minimálně jednoho statistika, který tuto taktickou přípravu zajišťuje. Obvyklá data, která se zaznamenávají, jsou následující [21]:

- počet všech útoků
- počet úspěšných útoků vedoucích k přímému bodu
- počet chyb na útoku
- úspěšnost na útoku, která je definována jako $\frac{\#úspěšných\ útoků - \#chyb\ na\ útoku}{\#útoků}$, kde # označuje počet

- počet bloků vedoucích k přímému bodu
- počet kontrolovaných bloků, tzn. počet doteků bloku
- úspěšnost příjmu
- počet úspěšných servisů, neboli es
- počet chybných servisů

Žádná z těchto statistik neobsahuje informaci o poloze hráče ani míče. Statistiky od Data Volley[2] nebo Beach-Data[3] přidávají informaci o pozici. Zaznamenává se poloha hráče při každé jeho akci – příjem, nahrávka, útok – a také míče, kam byl odehrán. Poloha je zakódována zónou, ve které byla akce provedena/skončena. Informace zahrnuje také způsob provedení akce – například typ servisu, způsob nahrávky nebo kvalitu provedení přihrávky. Obě platformy umožňují také integraci videa. Výše zmíněné platformy umožňují i integraci videa. Detailní porovnání software a platforem pro zpracování zápasů je v sekci 1.1.

Proces získávání statistik je v dnešní době ruční práce. Statistik zaznamenává data v reálném čase v průběhu zápasu nebo zpětně na základě videa pomocí zvolené platformy. Sportovci pak tyto statistiky používají při taktické přípravě na jednotlivé zápasy, vlastní analýzu výkonu a určení priorit v další tréninkové fázi.

2.2 Algoritmy pro zpracování časových řad

▶ **Definice 2.2** (Časová řada)**.** *Časová řada je pole hodnot získaných v konkrétních časových okamžicích.*

Hodnoty v časových řadách bývají často zatížené šumem, které je potřeba před dalšími operaci odstranit [22]. K tomuto účelu se využívá vyhlazení časové řady například pomocí klouzavého průměru [23] nebo filtrací 1D signálu mediánem [24, 25].

2.2.1 Klouzavý průměr

Klouzavý průměr je jednoduchou metodou, která konstruuje novou časovou řadu, jejíchž hodnoty jsou vypočteny průměrem okolních hodnot původní časové řady [23]. Počet okolních hodnot je hyperparametrem metody. V případě vyhlazování se používá velikost $m = 2k + 1$, kde k označuje tzv. poloměr šířky okna, pod kterým se průměr počítá. Klouzavý průměr s šířkou okna m se značí m-MA. Matematicky lze výpočet vyjádřit následovně:

$$y_t = \frac{1}{2k+1} \sum_{j=-k}^{k} x_{t+j}, \quad t = k+1, k+2, \ldots, n-k \tag{2.1}$$

[2] https://www.dataproject.com/Products/EU/en/Volleyball/DataVolley4
[3] https://www.beach-data.com

2.2.2 Mediánový filtr

Mediánový filtr je často aplikován na obrazová data, ale jeho jednodimenzionální verze lze aplikovat i na časovou řadu. Algoritmus využívá opět pohybujícího okna fixní velikosti, kde hodnota ležící na středu okna je nahrazena mediánem v rámci celého okna. Hlavní výhodou mediánového filtru je zachování hran, ale zároveň odstranění jedinečných odlehlých hodnot [24].

$$y_t = \text{median}_{j\in[-k,k]}\{x_{t+j}\} \quad t = k+1, k+2, \ldots, n-k \tag{2.2}$$

2.2.3 Rozklad časové řady

V časových řadách se popisuje několik vlastností – trend, sezónnost a cyklické změny [26]. Trend existuje, pokud je dlouhodobý nárůst/pokles hodnot časové řady. Sezónnost je vlastnost periodicky se opakujícího vývoje časové řady se známou fixní periodou. Cyklické změny jsou fluktuace, které nemají fixní periodu.

Časové řady lze rozložit pomocí dvou modelů, aditivního a multiplikativního. Aditivní model vyjádřen pomocí rovnice 2.3 je vhodný pro časové řady, kde amplituda sezónní složky je přibližně stále stejná v čase. Multiplikativní model vyjádřen rovnicí 2.4 je vhodným, pokud je sezónnost ovlivněna růstem/poklesem trendu.

$$y_t = S_t + T_t + R_t \tag{2.3}$$

$$y_t = S_t \times T_t \times R_t \tag{2.4}$$

Y_t vyjadřuje pozorovanou veličinu v čase t, S_t sezónní složku s periodou m, T_t trend obsahující i cyklickou změnu a R_t reziduum, které není vysvětlitelné, typicky jde o nějaký šum [26].

Dekompozice na aditivní model je prováděna pomocí metody klouzavého průměru. Konkrétně se jedná čtyř-krokový algoritmus:

1. T_t složka je vypočítaná aplikací klouzavého průměru s oknem o velikosti m, který je následován klouzavým průměrem s oknem o velikosti 2, kde m je sudé číslo. V lichém případě je použita pouze jedna aplikace m-MA.
2. Detrendování časové řady $y_t - T_t$.
3. Sezónní složka pro každou sezónu je vypočtena jako průměr detrendované časové řady v dané sezóně a v sumě se sčítají na 0.
4. Reziduum je zbytek po odečtení trendu a sezónnosti, konkrétně $R_t = y_t - T_t - S_t$.

2.3 Zpracování obrazu

Metody zpracování obrazu slouží k zisku informací, potlačení užitečných informací nebo naopak utlumení informací neužitečných ze snímku. Kapitola je zaměřena na algoritmy použité v analytické a praktické části, konkrétně se jedná algoritmus optical flow a metriku podobnosti obrazů.

2.3.1 SSIM index

SSIM, neboli index strukturální podobnosti, je metrikou, která porovnává dva signály – typicky referenční obraz s obrazem zkresleným [27]. SSIM se skládá ze tří komponent: svítivost, kontrast a struktura. Svítivost je funkcí střední hodnoty (μ_x) jasu. Kontrast obrazu je aproximován směrodatnou odchylkou (σ_x). Porovnání svítivosti dvou signálů $\mathbf{x}, \mathbf{y}$ je definováno

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \tag{2.5}$$

kde C_1 je konstantou zaručující nedělení nulou. Porovnání kontrastu dvou signálů využívá stejného principu

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \tag{2.6}$$

kde C_2 plní stejnou funkci jako konstanta C_1. Obě tyto konstanty jsou závislé na dynamickém rozsahu obrazu a $L << 1$ vztahem

$$C_{1,2} = (K_{1,2}L)^2 . \tag{2.7}$$

Strukturní porovnání je prováděno na základě korelačního koeficientu

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}. \tag{2.8}$$

Výsledný SSIM index daného signálu je násobkem všech tří komponent

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^\alpha \cdot [c(\mathbf{x}, \mathbf{y})]^\beta \cdot [s(\mathbf{x}, \mathbf{y})]^\gamma. \tag{2.9}$$

SSIM index je počítán lokálně pod oknem velikosti 8×8, které se posouvá po obrazu po pixelech. Globální SSIM index je průměrem všech lokálních výpočtů a vyjadřuje míru podobnosti dvou snímků. Metoda je implementována v knihovně `scikit-image`[4]

2.3.2 Optical flow – Farnerbackova metoda

Optical flow v obraze reprezentuje vzorec pohybu objektů a hran dvou po sobě jdoucích snímků, který je vyjádřen vektorem pohybu včetně rychlosti [28]. Výpočet optical flow předpokládá konstantní jas pozorovaných objektů a stejný vzorec pohybu sousedních bodů, což je na reálných snímcích těžko dosažitelné. Optical flow je možné dělit na dva typy: sparse a dense optical flow. Sparse optical flow je vypočítáván pouze na vybraných objektech v obraze, například na hranách či rozích. Naopak dense optical flow je vyjádřen pro všechny pixely v obraze.

Základními algoritmy pro výpočet optical flow jsou Horn-Schuck metoda [29], Lucas-Kanade metoda [30] a Farnebackova metoda [31]. Optical flow se také stal součástí oblasti hlubokého učení a existují přístupy využívající modely neuronových sítích, např. DeepFlow [32] či PWC-Net [33].

Myšlenkou Farnerbackova algoritmu je aproximovat nějaké okolí každého pixelu obrázku pomocí kvadratického polynomu, kde $\mathbf{A}$ je symetrická matice, $\mathbf{b}$ je vektor a c skalár [31].

$$f(\mathbf{x}) \sim \mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c \tag{2.10}$$

[4] https://scikit-image.org

První obrázek je vyjádřen pomocí polynomu f_1 a druhý obrázek pomocí polynomu f_2. Translace $\mathbf{d}$ lze vyjádřit následující rovnicí a má řešení, pokud $\mathbf{A_1}$ není singulární.

$$\mathbf{d} = -\frac{1}{2}\mathbf{A}_1^{-1}(\mathbf{b}_2 - \mathbf{b}_1) \tag{2.11}$$

V teoretickém prostředí by se $\mathbf{A_1} = \mathbf{A_2}$, ale v reálném prostředí se $\mathbf{A}$ aproximuje průměrem $\mathbf{A_1}, \mathbf{A_2}$, stejně tak $\mathbf{b_1}, \mathbf{b_2} \rightarrow \mathbf{\Delta b}$. Výsledný posun je vypočítán přes celé okolí pixelu za předpokladu, že posunutí se mění pomalu (rovnice 2.12), kde $w(\mathbf{\Delta})$ značí váhovou funkci pro body okolí. Právě tento předpoklad je nevýhodou navrhovaného algoritmu.

$$\sum_{\mathbf{\Delta x} \in I} w(\mathbf{\Delta x}) \| \mathbf{A}(\mathbf{x} + \mathbf{\Delta x})\mathbf{d}(\mathbf{x}) - \mathbf{\Delta b}(\mathbf{x} + \Delta \mathbf{x}) \|^2 \tag{2.12}$$

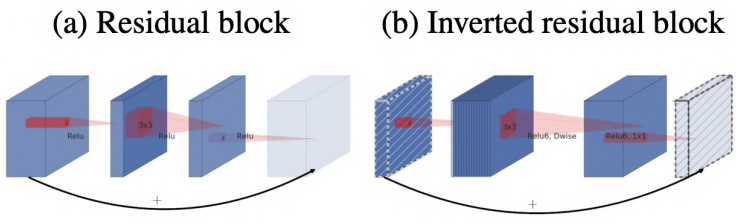
Výsledný algoritmus je iterativní a neporovnávají se posuny na stejné pozici, ale je využit apriori přístup, kde se porovnává $\mathbf{x}$ s $\mathbf{x} + \tilde{\mathbf{d}}(\mathbf{x})$.

2.4 Hluboké učení

Strojové učení je jedním z nejpopulárnějších a nejvíc rozšiřujících se odvětví technologie. Odvětvím strojového učení je hluboké učení využívající neuronových síti, které byly inspirovány schopností lidského mozku učit se. Neuronové sítě se používají k úlohám rozpoznání obrazu, segmentace, čtení textu, překlad slov, rozpoznání řeči a mnoho dalších. Právě pro úlohy související s obrazem jsou využívány transformery jako state-of-the-art a konvoluční sítě [34].

2.4.1 Mobile inverted bottleneck konvoluce

Sandler a kolektiv v roce 2019 navrhli model MobilNetV2 pro mobilní aplikace za účelem zachování efektivity a snížení výpočetní náročnosti [35]. Hlavními stavebními kameny jsou inverzní reziduální blok a separabilní konvoluce. Inverzní reziduální blok na rozdíl od klasického reziduálního bloku propojuje místa bottlenecku – místy s nejméně kanály (viz. Obrázek 2.2).



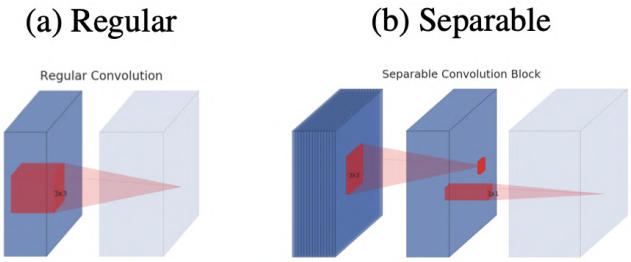
Obrázek 2.2 Rozdíl mezi reziduálním a inverzním blokem [35].

Separabilní konvoluce je složena z dvou operací – depth-wise konvoluce a point-wise konvoluce. Depth-wise konvoluce znamená aplikaci konvolučního filtru o velikosti $k \times k$ na každém

Tabulka 2.1 Architektura EfficientNet-B0 [36].

Fáze i	Operátor $\hat{\mathcal{F}}_i$	Rozlišení $\hat{H}_i \times \hat{W}_i$	#Kanálů $\hat{C}_i$	#Vrstev $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

kanálu zvlášť. Point-wise konvoluce je standardní konvolucí přes všechny kanály ale s kernelem o velikosti 1×1, tudíž se jedná o lineární kombinace skrz všechny kanály nezávisle pro každou pozici. Rozdíl mezi klasickou konvolucí a separabilní konvolucí je zobrazen na následujícím obrázku 2.3. Použití separabilní konvoluce snižuje výpočetní složitost téměř k^2-krát, kde k značí velikost kernelu.



Obrázek 2.3 Rozdíl mezi konvolucí a separabilní konvolucí [35].

Mobile inverted bottleneck konvoluce je spojením inverzního reziduálního bloku a separabilní konvoluce. Skládá se ze tří vrstev a reziduálního spojení před první vrstvou a po poslední vrstvě:

1. Conv1x1, BN, Relu – výstup má zvětšenou dimenzi počtu kanálů.
2. Depth-wise Conv3x3, BN, Relu – počet kanálů zůstává stejný z předchozí vrstvy.
3. Conv1x1, BN – redukce počtu kanálů na původní.

2.4.2 EfficientNet

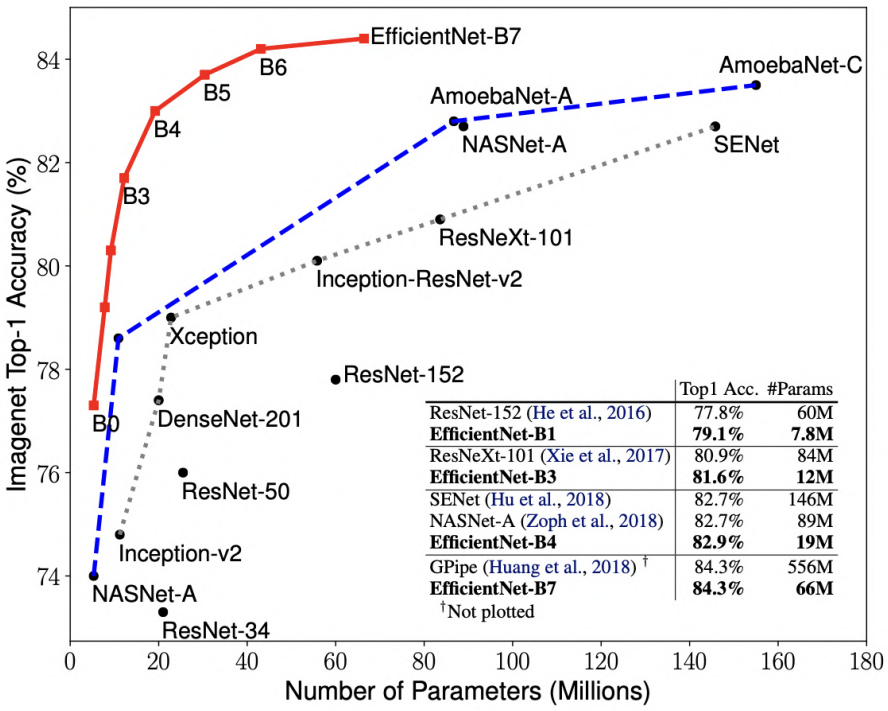
V roce 2020 M. Tan a Q.V. Le představili novou architekturu EfficientNet, která je optimalizovaná pro dosažení vysoké přesnosti při zachování malého množství parametrů a nároků na výpočetní výkon modelu. Spolu s architekturou navrhli metodu složeného škálování sítě v hloubce, šířce a rozlišení zároveň [36]. Architektura baseline modelu EfficientNet-B0 byla vytvořena pomocí multi-objektového neuronového prohledávání NAS. Stavebními bloky je klasická a separabilní konvoluce (viz. Tabulka 2.1)

Škálování hloubky sítě zvyšuje počet vrstev sítě s myšlenkou: čím hlubší síť, tím komplexnější zachycení příznaků. Škálování šířky sítě znamená zvětšování počtu kanálů a škálování rozlišení

zvětšuje rozlišení vstupních dat. Složené škálování škáluje všechny tři složky dle následující rovnice 2.13, kde ϕ je koeficient škálování.

$$\begin{aligned} \text{hloubka: } d &= \alpha^\phi \\ \text{šířka: } w &= \beta^\phi \\ \text{rozlišení: } r &= \gamma^\phi \\ \text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\ \alpha \geq 1, \beta &\geq 1, \gamma \geq 1 \end{aligned} \tag{2.13}$$

Pro zdvojení komplexity modelu autoři nalezli nejlepší trade-off mezi počtem parametrů a přesnosti hodnoty parametrů $(\alpha, \beta, \gamma) = (1.2, 1.1, 1.15)$, podle kterých byly škálovány ostatní modely rodiny EfficientNet B1-B7. Porovnání EfficientNet modelů s ostatními konvolučními modely na datasetu ImageNet je zobrazeno na Obrázku 2.4.



Obrázek 2.4 Porovnání EfficientNet modelů – přesnost vs. počet parametrů [36].

2.4.3 EfficientNetV2

O rok později v roce 2021, stejní autoři představili novou verzi EfficientNet nazvanou EfficientNetV2, která se zaměřuje na zkrácení času tréninku a zlepšení efektivity jednotlivých parametrů sítě [37]. Nová architektura vychází z bottlenecků architektury předchozí verze a primárně reaguje na vysoký čas tréninku při velkém rozlišení vstupních dat. Depth-wise konvoluce v počáteční fázi zpomaluje výpočet, a uniformita při škálování není sub-optimální. Pro zrychlení výpočtu jsou separabilní konvoluce v počátečních vrstvách nahrazeny konvolucí 3×3. Architektura baseline EfficientNet-S je popsána v Tabulce 2.2. Škálováním byly vytvořeny modely M, L. EfficientNetV2 se v porovnání k EfficientNetu trénuje 11-krát rychleji a zároveň je 6,8-krát menší.

Tabulka 2.2 Architektura EfficientNet-S [36].

Fáze	Operátor	Krok	#Kanálů	#Vrstev
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

2.4.4 Label smoothing

Label smoothing je technika regularizace používaná při tréninku neuronové sítě [38]. Cílem je zlepšit generalizaci modelu tím, že mírně modifikuje labely ve vstupních datech. Účelem metody je zabránit modelu být si příliš jistý predikcí majoritní třídy, lépe generalizovat na nových datech a být více odolný proti šumu.

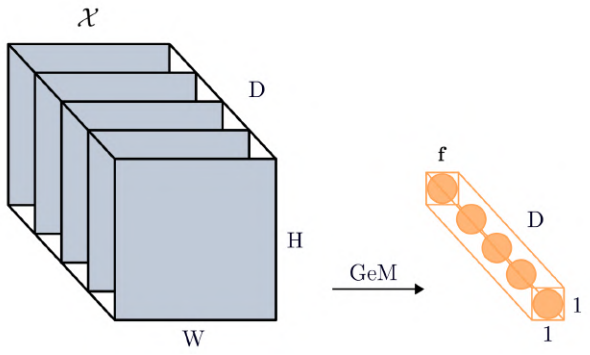
$$(1-\alpha)\mathbf{1}_{\text{gold}} + \alpha\frac{1}{\text{počet tříd}} \tag{2.14}$$

2.4.5 GeM pooling

GeM, neboli Generalized mean pooling je technika globálního poolingu kombinující max-pooling a average-pooling [39]. Metoda vypočítává obecný průměr přes všechny kanály vstupu. Vstupem je 3D tenzor $\mathcal{X}$ o rozměrech $W \times H \times D$, kde D určuje počet kanálů dle použité architektury. $\mathcal{X}_d$ vyjadřuje 2D vektor příznaků kanálu d. Výstup je vektor definován rovnicí

$$\mathbf{f}^{(g)} = \left[\mathrm{f}_1^{(g)} \dots \mathrm{f}_d^{(g)} \dots \mathrm{f}_D^{(g)}\right]^\top, \quad \mathrm{f}_d^{(g)} = \left(\frac{1}{|\mathcal{X}_d|}\sum_{x\in\mathcal{X}_d} x^{p_d}\right)^{\frac{1}{p_d}}. \tag{2.15}$$

Pooling parametr p_d určuje míru mezi average a max-poolingem. GeM je speciálním případem average-poolingu při volbě $p_d = 1$, max-poolingu při $p_d \to \infty$. Parametr p_d může být zvolen ručně či být naučen díky diferencovatelnosti funkce.



Obrázek 2.5 GeM pooling.

Kapitola 3
Analýza

Jedna beachvolejbalová sezóna zahrnuje více než 1700 turnajů světové série. Průměrná doba jednoho zápasu je 46 minut. Celková stopáž všech záznamů zápasů přesahuje 78 200 minut, což odpovídá 1 303 hodinám. Manuální nastříhání jednoho zápasu na jednotlivé výměny zabere přibližně 15 minut času. Zpracování všech zápasů jedné sezóny by trvalo přibližně 400 hodin a jedná se o čistě manuální práci bez nutnosti větší znalosti problematiky beachvolejbalu.

Automatizace střihu videa zefektivní proces tvoření statistických analýz a umožní skautům se zaměřit na odbornější úkony.

3.1 Data

Videozáznamy beachvolejbalových zápasů ze sezóny 2023 byly získány z oficiální databáze světové volejbalové federace FIVB. Oficiální turnaje pořádané FIVB na světové úrovni se rozdělují do tří kategorií: **Future, Challenger, Elite16**.

Kategorie Elite16 představuje nejvyšší úroveň soutěže pro 16 elitních týmů. O čtyři postupová místa do hlavní soutěže bojuje 16 týmů v kvalifikaci, která se odehrává podle systému na jednu prohru. Zápasy se konají na dvou kurtech – centrálním (CC) a vedlejším (C2). Jeden turnaj kategorie Elite16 zahrnuje 12 kvalifikačních zápasů a 36 zápasů v hlavní soutěži. Většina turnajů se koná pro muže i ženy zároveň, což může zdvojnásobit celkový počet zápasů na 96.

Turnaje v rámci kategorie Challenger představují příležitost pro širší spektrum týmů a patří do prostřední kategorie podle bodového hodnocení a finančních odměn. Kvalifikací postupuje 8 týmů ze 32 do hlavní soutěže. Hlavní soutěž se pořádá pro 24 dvojic. Vzhledem k velkému množství zápasů při pořádání turnaje pro obě kategorie zároveň, se odehrávají zápasy až na 6 kurtech (CC, C2-C6). Celkový počet zápasů na jeden turnaj kategorie Challenger činí 24 zápasů v kvalifikační fázi a 42 zápasů ve fázi hlavní soutěže.

Kategorie Future je poslední kategorií světových turnajů a nachází se konci bodového ohodnocení. Turnaje jsou obsazovány spíše nově sestavenými či mladými týmy, které nemají v aktuální sezóně olympijské ambice. Kvalifikace je určena pro 16 týmů se čtyřmi postupovými místy. Hlavní soutěž má kapacitu též 16 týmů. Jeden turnaj kategorie Future zahrnuje celkem 40 zápasů, z toho 12 v kvalifikační fázi a 28 ve fázi hlavní soutěže.

Zaznamenávání zápasů ze všech kurtů je povinné pro turnaje kategorie Elite16 a Challenger.

Každý kurt snímá minimálně jedna kamera, která zabírá celou plochu hrací zóny. Úhel snímaní a perspektivní transformace kurtu jsou unikátní pro každý kurt, jelikož není obecně definována snímací soustava, jak by měl být kurt snímán (viz. Obrázek 3.1). Všechny kamery mají společné rozlišení **1920 × 1080 pixelů** a snímkovací frekvenci **50 snímků za sekundu**. Záběr z jedné kamery se během zápasu nemění a je konstantní bez střihů. Turnaje typu Future mají povinnost vysílat zápasy pouze finálových dnů.

(a) Vedlejší kurt C2 – Elite16 Paříž.

(b) Centrální kurt CC – Challenger Goa.

Obrázek 3.1 Odlišné snímací úhly.

3.1.1 Kvalita videa

Kvalitu videa určuje nejen nastavení snímacího úhlu kamery, ale především světelné podmínky, které jsou velmi ovlivněny počasím a časem, kdy se zápas odehrává. Z toho důvodu byla subjektivně ohodnocena kvalita 475 záznamů z turnajů typu Elite16. Škála kvality byla pro jednoduchost hodnocení stanovena na tříprvkovou množinu $\{1, 2, 3\}$. Vlastnosti obrazu, na které byl kladen důraz, jsou následující:

- Zaostření na kurt a hráče
- Vyváženost histogramu
- Kontrast hráčů vůči písku
- Stíny hráčů a okolí na hřišti
- Textura písku – při špatném počasí jsou prohlubně v písku více viditelné a vytvářejí tmavé mapy
- Kvalita umělého osvětlení, pokud se jedná o noční zápas
- Klepání kamery

Známka kvality 1 značí výborné jasové podmínky. Kurt a hráči jsou zaostřeni a kontrastní vůči písku. Zároveň textura písku je hladká a nenachází se zde mnoho stínů. Videa takto ohodnocena mají velký potenciál k úspěšnému automatickému zpracování. Videa ohodnocena známkou 2 trpí nějakou nedokonalostí z vyjmenovaných vlastností, zejména se jedná o vyšší koncentraci stínů na hřišti, malý kontrast hráče vůči písku či nevyvážený histogram. Videa ohodnocená známkou 3 mají velmi nevyvážený histogram a ve videu převažují extra světlé nebo extra tmavé barvy, kontrast hráčů s hřištěm je nízký kvůli vysoce texturovanému terénu písku. Špatná kvalita videa pravděpodobně neumožní úspěšné automatické zpracování. Srovnání kvality videí a jejich ohodnocení je zobrazeno na následujícím obrázku (Obrázek 3.2).

(a) Ohodnocení: 1.

(b) Ohodnocení: 2, výrazná textura písku.

(c) Ohodnocení 2, výrazné stíny na kurtu.

(d) Ohodnocení: 3, velmi tmavé.

Obrázek 3.2 Porovnání kvality snímků dle hodnocení.

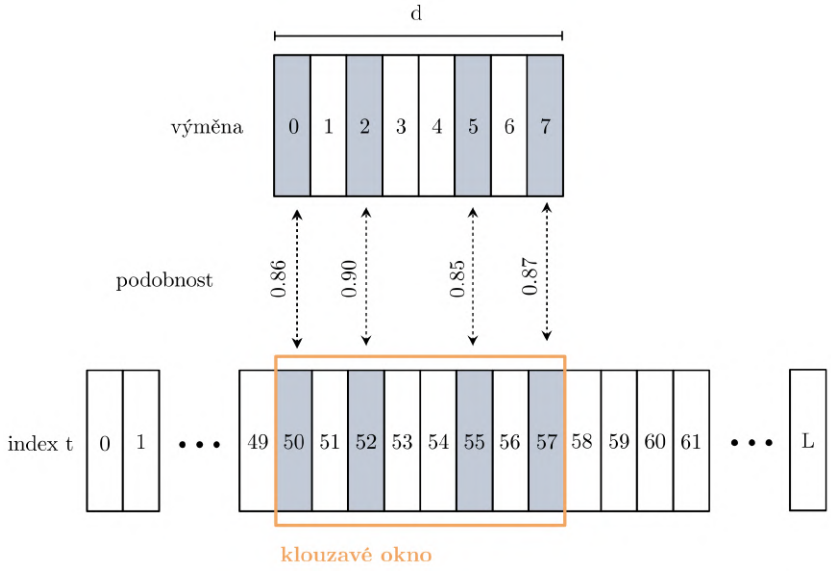
Celkově bylo ohodnoceno 475 záznamů ze sedmi různých turnajů kategorie Elite16. Známku 1 obdrželo 212 videí (44,7 %), o stupeň horší známku obdrželo 222 videí (46,7 %) a v nejhorší kategorii kvality je pouze 41 videí (8,6 %), kde se primárně jedná o jeden konkrétní turnaj – Elite16 v Montrealu. Videa ohodnocená nejhorší známkou kvality nebyla používána v dalších fází práce.

3.1.2 Skutečná data

Skutečná data pro automatický střih ve tvaru časových značek byla získána dvěma způsoby ve spolupráci s firmou BEACH – DATA s.r.o a jejich produktu Beach Data Video[1]. Proces získávání dat je popsán v kapitole 1.1.2. Střihání videozáznamu z FIVB databáze daného zápasu probíhá již po naskatouvání zápasu. Beach Data definují výměnu od zapískání rozhodčího po dotek míče s kurtem či ukončení výměny rozhodčím. V běžné praxi skauti video stříhají až ve čtyřnásobné rychlosti, tudíž je automaticky zanesena lehká nepřesnost. Beach Data ukládá informace o výměnách pro každý naskautovaný zápas dvěma způsoby. Prvním způsobem je ukládání vystřižených výměn ve formátu jednotlivých videí, tzn. že každému zápasu náleží sada nastříhaných výměn z původního videa v rozlišení 640×360 pixelů. Během podzimu 2023 byl přidán druhý způsob uložení informace o střihu, a to formou časových značek. Ke každé nastříhané výměně zápasu je uloženo jak video, tak časová značka, kde se výměna nachází vůči původnímu záznamu, ze které byla výměna extrahována. Časová značka jednotlivé výměny je ve formátu dvojice, kde první prvek definuje sekundu, ve které výměna začala a druhý prvek sekundu, ve které výměna skončila. Seznam časových značek určuje skutečná data všech výměn daného zápasu.

[1] https://www.beach-data.com

Dále jsou v práci využívána skutečná data pouze ve formátu seznamu časových značek. Pro zisk časových značek z již nastříhaných výměn byl implementován algoritmus využívající klouzavé okno. Principem algoritmu je postupné prohledávání původního videa za účelem nalezení shody s danou výměnou. Výměny jsou seřazeny, tudíž se pro každou následující výměnu prohledává menší část videa. Z každého videa výměny jsou extrahovány čtyři snímky – začáteční, v první čtvrtině, ve třech čtvrtinách a poslední snímek. Klouzavé okno se po původním videu posouvá skokem po padesáti snímcích, což vede na snímkovací frekvenci jeden snímek za sekundu. Video je zmenšeno na stejné rozlišení jako záznamy výměn, 640×360 pixelů. Pro každý okamžik t původního videa jsou extrahovány snímky na pozicích $t, t + \frac{1}{2}d, t + \frac{3}{4}d, t + d - 1$, kde d značí délku hledané výměny v sekundách (viz. Obrázek 3.3). Čtyři snímky výměny a čtyři snímky původního videa jsou porovnány a ohodnoceny metrikou podobnosti SSIM (`scikit-structural_similarity`[2]). Pokud průměr metriky podobnosti přes všechny čtyři snímky je větší než $0,85$, je výměna považována za nalezenou. V opačném případě je původní video posunuto na index $t+50$ a proces je opakován, dokud není nalezena shoda nebo původní video není u konce (délka původního videa je značena písmenem L).



Obrázek 3.3 Algoritmus klouzavého okna pro nalezení časové značky výměny.

Součástí zisku skutečných dat je i obdržení informace o počtu výměn v každém setu a zda-li je výměna odehrána po oddechovém čase, který byl zažádán jedním z týmů.

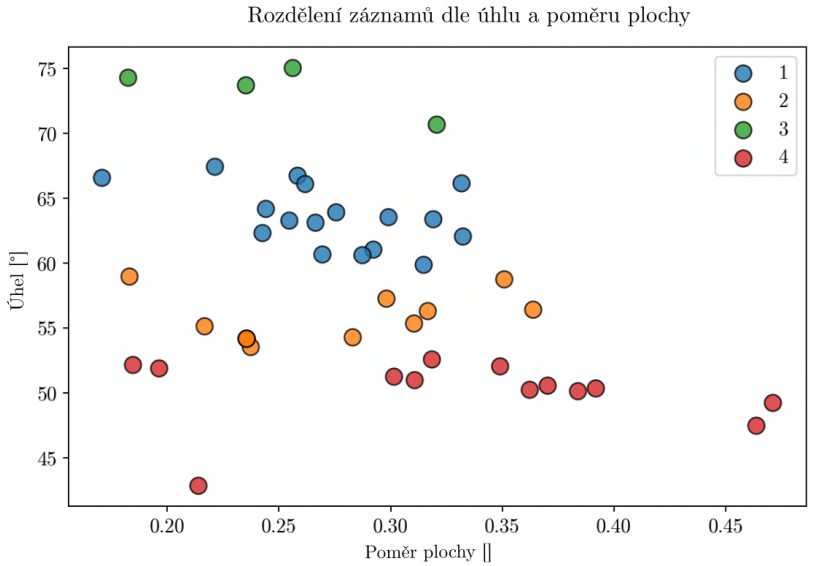
3.1.3 Sestavení datasetů

V sezóně 2023 bylo odehráno 9 turnajů kategorie Elite16, 10 turnajů kategorie Challenger a 1 finálový turnaj pro prvních deset týmů světového žebříčku. Pro každý hrací kurt turnaje byly ručně získány body v obrazu, které odpovídají rohům hřiště. Skutečná data sezóny 2023 byla nasbírána pro celkový počet 324 zápasů, což odpovídá zhruba 250 hodinám záznamu. Pro testovací účely byly přidány ještě záznamy dvou turnajů ze začátku sezóny 2024 – Challenger Recife, Elite16 Doha, které čítají dohromady 88 zápasů.

[2]https://scikit-image.org

Pro zrychlení práce s daty byl z videa extrahován každý desátý snímek a uložen ve formátu JPEG s rozlišením 1920 × 1080 pixelů. Extrakce probíhala pomocí `ffmpeg` frameworku[3]. Výsledná snímkovací frekvence videa uloženého stylem sady jednotlivých obrázků je pět snímků za sekundu. Uložení sekundy videa v dané snímkovací frekvenci trvá průměrně 0,1 sekundy.

Kvůli velkému objemu dat byly záznamy rozděleny do čtyř skupin pro rychlejší vyhodnocování experimentů. Záznamy byly rozděleny algoritmem K-means na základě dvou kritérií. Obě kritéria vycházejí ze znalosti, kde se nachází hřiště v obrazu. První kritérium je úhel svíraný v pravém dolním rohu hřiště. Druhým kritériem je poměr plochy hřiště vůči celému obrazu. Tyto dvě metriky v sobě nesou zjednodušenou informaci o perspektivním zkreslení obrazu. Výsledek algoritmu K-means je zobrazen na následujícím grafu (Obrázek 3.4). Jednotlivé body grafu představují odlišné druhy záznamů dle turnaje a kurtu, kde byl záznam pořízen. Konkrétní rozdělení záznamů do skupin je zobrazeno v příloze v tabulkách A.1, A.2, A.3, A.4.



Obrázek 3.4 Vizualizace výsledku K-means algoritmu, rozdělení záznamů dle úhlu a poměru plochy.

Za účelem využití hlubokého učení pro automatický střih videa byly záznamy každé skupiny rozděleny na trénovací, validační a testovací množinu (Tabulka 3.1). Data byla náhodně rozdělena tak, aby ve validační a testovací množině byl alespoň jeden zástupce každého druhu záznamu v rámci jedné skupiny. Zároveň je v každé skupině minimálně jeden druh záznamu, který nebyl používán při tréninku a slouží pouze pro validaci/testování, jako simulace chování na nově příchozích datech.

3.2 Algoritmy střihu videa

Byly navrženy tři algoritmy dvou typů. První algoritmus pracuje nesupervizovaně a využívá charakteristik beachvolejbalu, zatímco zbývající dva algoritmy jsou založeny na principu hlubokého učení. Efektivita automatického střihu není hodnocena pouze na základě přesnosti detekce výměn, ale i podle přesnosti v celkovém počtu předpovězených výměn. Díky detailnímu charakteru skutečných dat je implementována kontrola výsledků pro správný počet výměn.

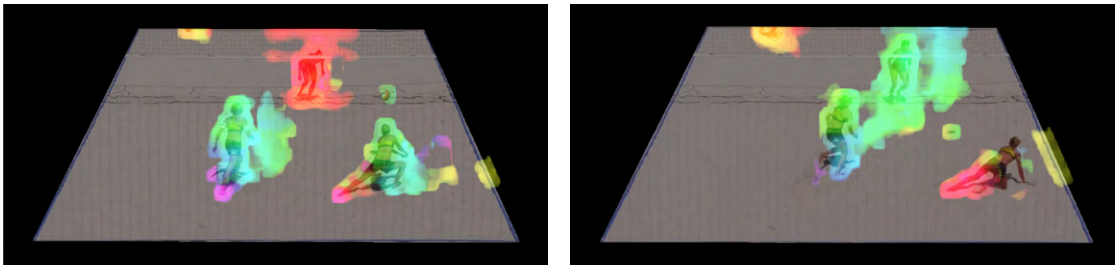
[3] https://ffmpeg.org

Tabulka 3.1 Velikosti datasetů dle skupiny.

Skupina	Trénovací data	Validační data	Testovací data
1	57	13	15
2	71	17	12
3	34	9	4
4	59	18	15

3.2.1 Optical Flow

Pro plážový volejbal je charakteristické, že krátce před zahájením výměny dojde na hřišti k dočasnému utišení, kdy se téměř nic nepohybuje. Hráči obvykle zaujmou své pozice před začátkem servisu, kde čekají na příjem nebo jsou připraveni na obranu. Jakmile je míč uveden do hry, scéna nabere na aktivitě, přičemž největší pohyb nastává krátce před ukončením výměny. I po skončení výměny se pohyb na hřišti vyskytuje, avšak v menší míře, kdy hráči oslavují získaný bod nebo se připravují na další výměnu. Tato charakteristická vlastnost je využita v prvním algoritmu střihu videa. Algoritmus je postaven na metodě optical flow, konkrétně na Farnerbackově metodě, která počítá pohyb každého pixelu mezi dvěma po sobě jdoucími snímky. Pohyb je reprezentován vektorem, kde první složka určuje směr a druhá velikost pohybového vektoru. Ve vizualizaci je směr pohybu znázorněn rozdílnou barvou a velikost vektoru intenzitou (Obrázek 5.1). Průměrem velikostí přes všechny pixely je vyjádřena tzv. dynamika scény.

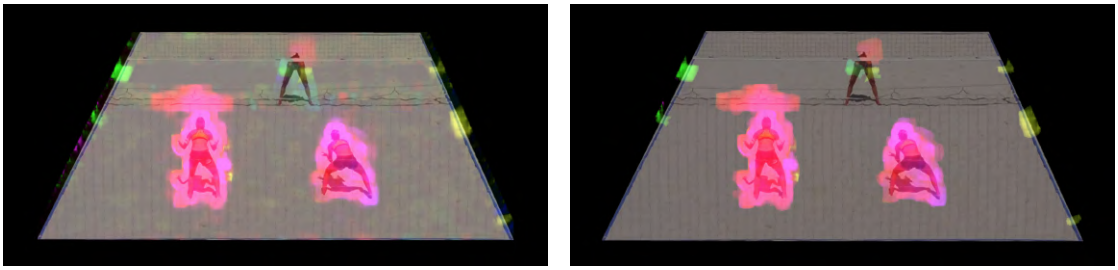


Obrázek 3.5 Vykreslení optical flow na snímku.

Algoritmus střihu videa pomocí optical flow se skládá z několika kroků. Prvním krokem je předzpracování videa, které zahrnuje vymaskování a oříznutí oblasti bez kurtu. Ponecháním pouze oblasti, kde se nachází kurt, dochází k eliminaci pohybu mimo hřiště – například pohyb podavačů míčů, povzbuzování fanoušků. Dalším krokem předzpracování je aplikace mediánového filtru k odstranění šum z obrazu. Po dokončení předzpracování se provádí výpočet optical flow pro každý snímek videa s požadovanou snímkovací frekvencí. Výstupem pro každý snímek je pole vektorů určující pohyb každého pixelu.

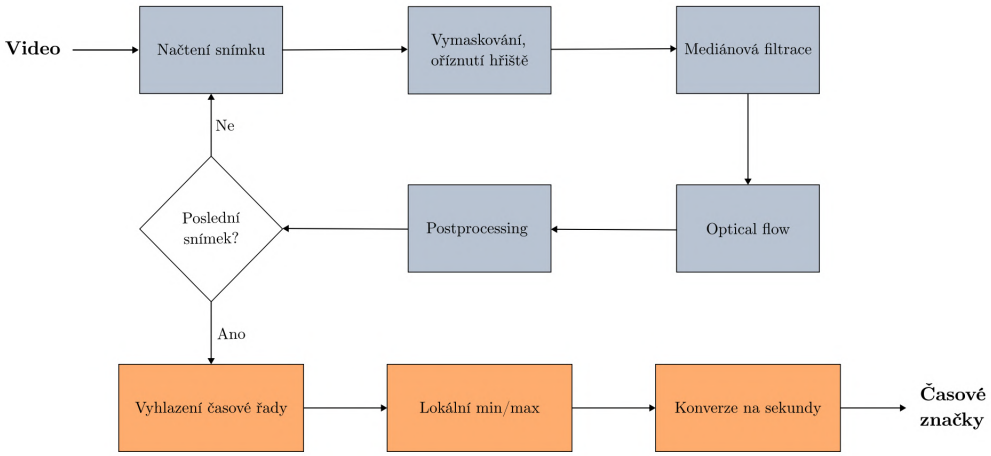
Před výpočtem celkové dynamiky snímku je aplikován postprocessing, který redukuje šum. Velikosti všech pohybových vektorů jsou normalizovány pomocí min-max normalizace na hodnoty v rozsahu 0 až 255. Celková dynamika snímku je ovlivněna pouze pixely s pohybem větším než 30 a zároveň jsou odstraněny shluky pixelů, které mají plochu menší než $\frac{1}{10000}$ plochy celého snímku (Obrázek 3.6). Odfiltrované pixely nejsou součástí výpočtu průměru velikostí pohybu, kterým je vyjádřena dynamika snímku. Výsledkem je časová řada, jež v každém okamžiku nese informaci o dynamice scény.

Dalším krokem je zpracování časové řady. Časová řada je nejprve normalizována min-max nor-



Obrázek 3.6 Porovnání dynamiky scény před filtrací (vlevo) a po filtraci (vpravo).

malizací a následně vyhlazena pomocí rozkladu časové řady, klouzavého průměru, mediánového filtru nebo kombinací z vyjmenovaných technik. Začátek výměny je detekován jako lokální minimum, konec výměny jako lokální maximum. Index lokálního minima a maxima je po konverzi na sekundy dle snímkovací frekvence časovou značkou začátku a konce výměny. Seznam časových značek všech detekovaných výměn definuje výstup algoritmu. Celkový proces je zobrazen na následujícím diagramu (Obrázek 3.7).



Obrázek 3.7 Diagram algoritmu střihu pomocí optical flow.

3.2.2 2D klasifikace

Automatický střih lze chápat jako problém supervizované klasifikace. Skutečná data, která byla získána, nesou informaci o každém snímku videa, zda je součástí výměny či ne. Problém automatického střihu je převeden na problém binární klasifikace. Pokud je snímek součástí výměny, je mu přiřazena třída `1`, v opačném případě třída `0`. Během tréninku se používá metoda label smoothingu na sousedící snímky výměny.

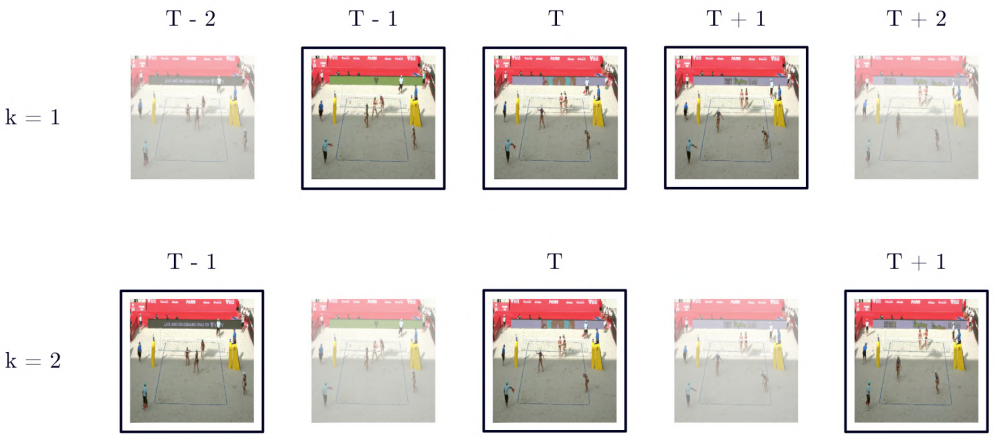
Video je posloupnost statických snímků, kde frekvence snímání zaručuje plynulost při přehrávání. Rozdíl mezi videem a jednotlivým snímkem spočívá pouze v časovém rozměru. Pokud je video rozloženo na jednotlivé snímky a je zanedbána časová souvislost mezi nimi, lze klasifikaci videa redukovat na klasifikaci jednotlivých snímků. Tento přístup využívá druhý algoritmus střihu videa na jednotlivé výměny.

Základním kamenem algoritmu je konvoluční hluboká neuronová síť typu EfficientNet. Data vstupující do modelu jsou augmentovaná pro větší regularizaci. Na výstup z EfficientNetu (2D enkodéru) je aplikován GeM pooling pro zisk vektoru příznaků, který vstupuje do klasifikační hlavy predikující pravděpodobnosti příslušnosti k třídě `1`. Pravděpodobnost příslušnosti k třídě `1` je predikovaná pro každý snímek videa dle specifikované snímkovací frekvence a vzniká tím opět časová řada, kde časovou osu tvoří indexy jednotlivých snímků. Časová řada je vyhlazena, zaokrouhlena a jsou vytvořeny intervaly po sobě jdoucích jedniček, které reprezentují časové značky výměn.

3.2.3 2.5D klasifikace

Klasifikace využívající 2D konvolučního filtru o hloubce $3N$ pro RGB snímky nebo hloubce N pro snímky šedotónové, kde N vyjadřuje počet spojených snímků, je dále nazývaný 2.5D klasifikací. Jedná se o rozšíření předchozího algoritmu 2D klasifikace. Vstupem do neuronové sítě je shluk N RGB snímků, kde N je liché číslo. Rozestupy po sobě jdoucích snímků jsou určeny parametrem kroku k (viz. Obrázek 3.8). Model predikuje, zda středový snímek T je součástí výměny na základě sousedních snímků z množiny (Obrázek 1.10). Shluky vytvořené z krajních snímků videa obsahují duplikáty posledního platného snímku, aby byla zachována velikost shluku N, například je-li predikce prováděna na prvním snímku videa, shluk obsahuje $\lceil N/2 \rceil$ středových snímků T a $\lfloor N/2 \rfloor$ snímků sousedících v posloupnosti $(T+1, \ldots, T+\lfloor N/2 \rfloor)$.

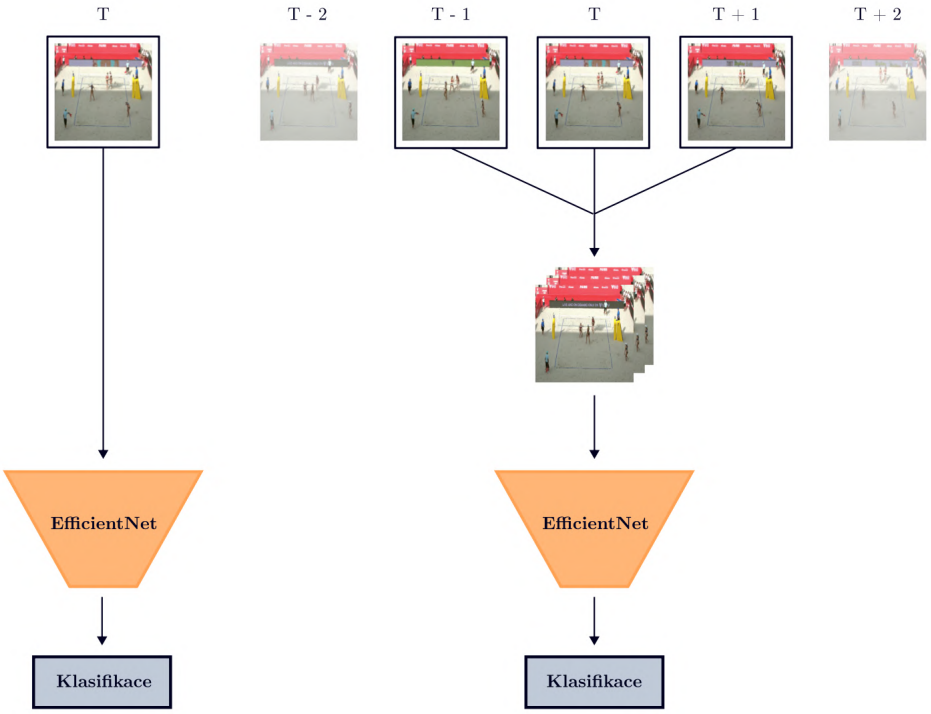
Architektura konvolučního modelu je založena na architektuře EfficientNet. Příznaky, které vystupují z modelu EfficientNet, slouží jako vstup (po aplikaci GeM poolingu) do klasifikační hlavy, která předpovídá příslušnost k třídě `1`. Tímto způsobem je vytvořena časová řada ze všech snímků, z které jsou extrahovány časové značky stejným způsobem jako v předchozím algoritmu. Porovnání architektur dvou algoritmů je zobrazeno následujícím obrázku (Obrázek 3.9).



Obrázek 3.8 Struktura vstupních dat do modelu 2.5D klasifikace.

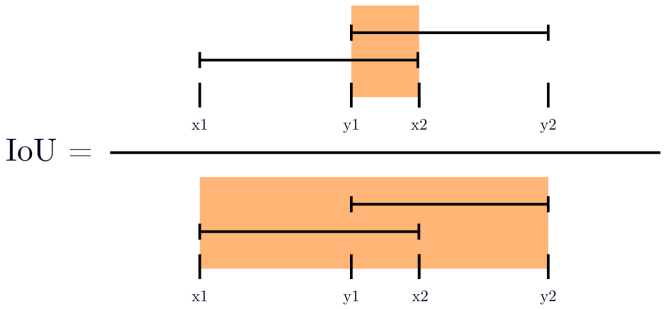
3.2.4 Metriky kvality střihu

Algoritmy jsou evaluovány metrikami pro klasifikaci a správnost střihu. Mezi základní metriky patří přesnost a F1 skóre. **Přesnost** (acc) je vyjádřena jako počet správně oklasifikovaných snímků ku celkovému počtu snímků ve videu. **F1 skóre** je zahrnuto do evaluace, jelikož snímků bez výměny je více než snímků obsahující výměny v poměru 2:1.



Obrázek 3.9 Porovnání architektur 2D (vlevo) a 2.5D klasifikace (vpravo).

Kvalita vystřižených výměn byla ohodnocena třemi metrikami: IoU, chyba začátku, chyba konce. **IoU**, neboli Intersection over Union, popisuje rozsah překrytí dvou obdélníků. čím víc se obdélníky překrývají, tím větší je hodnota. IoU je počítána jako poměr plochy průniku ku ploše sjednocení (3.10). Obdélníky jsou reprezentovány intervalem začátku a konce výměny z časových značek. **Chyba začátku** (err start) výměny je průměrnou absolutní chybou v sekundách, o kolik se začátek liší vůči pravdivým datům. **Chyba konce** (err end) výměny je analogicky průměrnou absolutní chybou v sekundách, o kolik se liší konec výměn. Každý predikovaný interval je spárován s takovými skutečnými intervaly, se kterými mají nenulový průnik. Může nastat situace, kdy predikovaný interval není spárován s žádným skutečným intervalem nebo naopak s více intervaly naráz.

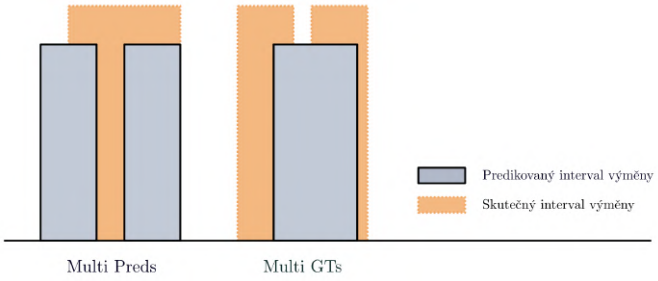


Obrázek 3.10 IoU – (x1, x2) predikovaný interval, (y1, y2) skutečný interval.

Hlavním cílem automatického střihu beachvolejbalových zápasů je dosáhnout co nejvyšší přesnosti v určení celkového počtu výměn. Při tréninku a evaluaci je monitorován počet navíc

vygenerovaných výměn, tzn. počet výměn, které nejsou spárované s žádnou skutečnou výměnou (**FP**), počet chybějících výměn – počet skutečných výměn, ke kterým není spárována žádná predikovaná výměna (**FN**). Počet výměn, které jsou spárované přesně s jednou skutečnou výměnou, je definován jako **TP**. Dále jsou sledovány případy, kdy skutečná výměna je spárována s několika predikovanými výměnami (**Multi Preds**), a případy, kdy predikovaná výměna je spárována s více než jednou skutečnou výměnou (**Multi GTs**) – Obrázek 3.11.

Recall vyjadřuje pravděpodobnost, že skutečná výměna je detekována, a je spočítán jako podíl správně detekovaných výměn (kdy je predikovaná výměna spárována pouze s jednou skutečnou výměnou) a celkového počtu skutečných výměn. **Precision** určuje pravděpodobnost, že predikovaná výměna je opravdovou výměnou, a je vyjádřena vzorcem TP/(TP + FP). Naopak false discovery rate (**FDR**) určuje pravděpodobnost, že vygenerovaná výměna je výměnou navíc. FDR se vypočítá jako podíl FP a celkového počtu predikovaných výměn.



Obrázek 3.11 Znázornění metrik multi preds a multi gts.

Při inferenci se hodnotí také úspěšnost na úrovni celého videa. Video je označeno jako správně nastříhané, pokud metriky střihu FP, FN jsou rovny nule, tzn. celkový počet skutečných výměn je roven počtu předpovězených výměn. Dále se pozorují případy, kdy algoritmus udělal přesně 1 chybu nebo přesně 2 chyby – součet FP + FN je roven danému číslu chyb. Pakliže nastane situace, kdy je skutečná výměna spárována s několika predikovanými výměnami (Multi Preds), je navýšen počet FP. V situaci, kdy je predikovaná výměna spárována s více než jednou skutečnou výměnou (Multi GTs), je navýšen počet FN.

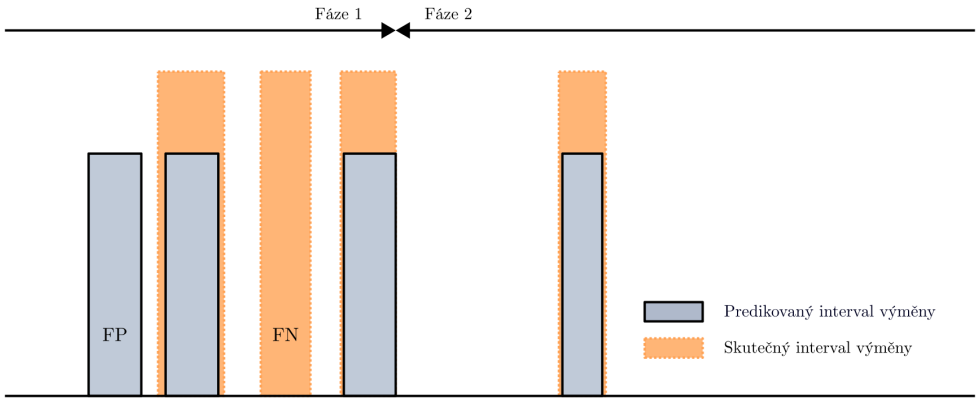
3.2.5 Kontrola výstupu časových značek

S ohledem na hlavní cíl automatického střihu zápasů beachvolejbalu – správné určení počtu výměn, byl implementován algoritmus, který sleduje správnost počtu výměn během jednotlivých částí zápasu. Algoritmus využívá dat, které jsou k dispozici i během inference. Jedná se o počet výměn v každém setu a indexy výměn, které proběhly po oddechovém čase. Zároveň dle pravidel beachvolejbalu nastává technický oddechový čas v prvních dvou setech hry po odehrání 21. výměny. Všechna tato data spojuje jedna vlastnost: po každé takové události následuje během zápasu pauza. Průměrná doba oddechového času je 71 sekund. Průměrná pauza mezi sety je 84 sekund a pauza mezi druhým a třetím rozhodovacím setem je prodloužena v průměru o 10 sekund.

Kontrola výstupu začíná rozfázováním zápasu na části, které jsou oddělené pauzou trvající déle než 60 sekund. Každá část je definována indexem výměny, se kterou fáze končí. Jedná se o indexy v rámci vypredikovaných výměn. Rozdíly v indexech nesou informaci o počtu výměn ve fázích. Počet predikovaných fází se může lišit od počtu fázích skutečných. Počet predikovaných fází je navýšen, pokud nastal nový míč, jeden z týmů požádal o přezkoumání rozsudku rozhodčího nebo si jeden z hráčů šel očistit brýle k rozhodčímu. Z toho důvodu jsou predikované fáze nama-

povány na fáze skutečné tak, aby rozdíl indexů koncových výměn predikované a skutečné fáze byl co nejmenší. V případě vyššího počtu skutečných fází oproti fázím z predikovaných intervalů – situace může nastat, pakliže během skutečné pauzy je vygenerována FP výměna – probíhá mapování skutečných fází na fáze predikované. Po namapování se od začátku do konce kontroluje rozdíl indexů, který určuje, jestli v dané fázi chybí, přebývá nebo je správný počet výměn. Při nenulovosti rozdílu je nalezená chyba propagována do dalších fází. Algoritmus v rámci jedné fáze nedokáže odhalit chybu, kdy je vygenerován správný počet výměn, ale na špatných pozicích, tzn. stejný počet FP výměn jako FN (viz. Obrázek 3.12).

Minimální rozdělení zápasu obsahuje 4 fáze v případě, že jde o dvousetový zápas a žádný z týmů nepožádá ani o jeden oddechový čas. Implementace kontroly umožňuje rychlejší manuální opravu automaticky detekovaných výměn, neboť je zmenšen objem videa, které je zapotřebí ručně zpracovat. Umožňuje také zpřesnění kontroly správného počtu výměn v jednotlivých fází, oproti pouhé kontrole rozdílu celkového počtu predikovaných výměn a skutečných výměn. Výstupem algoritmu je počet chybějících a počet přebývajících výměn v každé fázi zápasu.

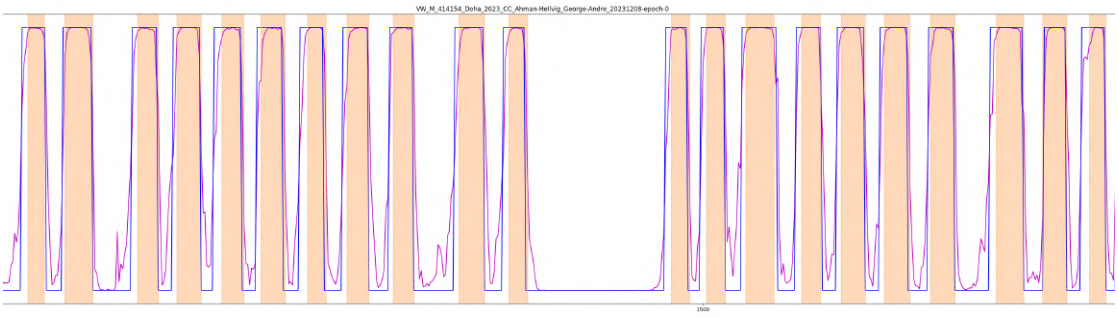


Obrázek 3.12 Situace neschopnosti kontroly výstupu časových značek.

Kapitola 4
Realizace

Algoritmy automatického střihu jsou implementovány v jazyce Python[1] s využitím knihoven OpenCV[2], Pandas[3] a Numpy [4]. Pro trénink a implementaci neuronové sítě je použita knihovna PyTorch[5]. Architektura neuronových sítí včetně předtrénovaných vah je získána z platformy Hugging Face[6]. Hugging Face je open source platformou pro podporu strojového učení. Obsahuje přes 350 tisíc modelů, 75 tisíc datasetů a 150 tisíc demo aplikací.

Pro vizualizace a ukládání experimentů jsou trénink a evaluace algoritmů napojeny na Neptune[7] – MLOps komponentu. Neptune umožňuje verzování souborů, real-time sledování využití paměti a výpisů na standardní výstup, porovnání experimentů mezi sebou a logování metrik. Pro snadnější porovnání a pochopení výsledků jsou po každé dotrénované epoše vygenerovány grafy výsledků každého videa na validační množině (Obrázek 4.1). Oranžové pruhy značí skutečné intervaly výměn, fialová křivka je výstup algoritmu – pravděpodobnost příslušnosti k třídě `1`, v případě algoritmu optical flow míra pohybu a modrá křivka je zpracovaný výstup modelu určující predikované výměny. X-ová osa je osou času v sekundách.



Obrázek 4.1 Ukázka vizualizace výsledků střihu videa.

[1] https://www.python.org
[2] https://opencv.org
[3] https://pandas.pydata.org
[4] https://numpy.org
[5] https://pytorch.org/
[6] https://huggingface.co
[7] https://neptune.ai

4.1 Optical Flow

Algoritmus střihu videa pomocí optical flow je postaven na Farnerbackově metodě, která je implementována v knihovně OpenCV – `cv2.calcOpticalFlowFarneback`. Pro rychlejší výpočet optical flow je vstupní video transformováno na menší rozlišení. Experimenty probíhaly s poměrem zmenšení z množiny $[\frac{1}{1}, \frac{1}{2}, \frac{1}{4}]$. Po vymaskování okolí hřiště je na snímek aplikován mediánový filtr s velikostí kernelu 5×5. Výpočet optical flow je spuštěn pro každý snímek s požadovanou snímkovací frekvencí. Snímkovací frekvence se v průběhu experimentů měnila od frekvence jednoho snímku za sekundu po pět snímků za sekundu.

Redukce šumu probíhá na úrovni každého snímku pomocí prahování a hledání kontur. Filtrace je aplikována na obraz pohybových vektorů (ve formátu HSV), kde odstín je definován směrem pohybového vektoru, jas určen velikostí vektoru a saturace nastavena na maximum. Velikosti pohybových vektorů v rámci jednoho snímku jsou normalizované min-max normalizací v rozsahu 0 až 255 a následně prahovány jedním prahem zespodu. Vyfiltrované vektory mají velikost menší než 30. Odstranění shluků je implementováno filtrací minimální plochy kontur nalezených v obrazu pohybových vektorů. Minimální plocha shluku vektorů je $\frac{1}{10000}$ plochy celého snímku. Výsledná dynamika snímku je průměrem velikostí všech pohybových vektorů, které prošly filtrací.

Časová řada vyjadřující dynamiku scény pro každý snímek je normalizována min-max normalizací a vyhlazena. Bylo testováno vyhlazení mediánovým filtrem, klouzavým průměrem, rozkladem časové řady a kombinace všech tří přístupů. Časová řada je rozložena dle aditivního modelu, kde amplituda sezónní složky odpovídá průměrné délce jedné výměny. Průměrná délka jedné výměny je z nasbíraných dat rovna 13 sekundám, nicméně v rámci experimentů byla amplituda nastavena od 10 do 14 sekund. Začátek výměny je detekován jako lokální minimum v okolí 10 sekund, konec výměny jako lokální maximum ve stejném okolí. Indexy lokálních minim/-maxim značí časovou značku dané výměny v jednotkách snímků, tudíž je provedena konverze na sekundy dle snímkovací frekvence použité při zisku optical flow.

Výpočty optical flow probíhají na CPU. Doba výpočtu se odvíjí od snímkovací frekvence a rozlišení. Při snímkovací frekvenci $fps = 1$ a čtvrtinovým rozlišením 480×270 pixelů trvá výpočet optical flow 15 % délky daného videa. Naopak při plném rozlišení 1920×1080 pixelů a snímkovací frekvenci $fps = 5$ je doba výpočtu téměř dlouhá jako originální video. Střední cestu představuje konfigurace poloviční rozlišení 960×540 pixelů se snímkovací frekvencí dva snímky za sekundu, kdy doba výpočtu trvá 22 % délky videa.

4.2 2D klasifikace

Modelem 2D klasifikace videa za účelem střihu je konvoluční neuronová síť s klasifikační hlavou. Konkrétně se jedná o modifikace architektury EfficientNetV2 – EfficientNetV2-S, EfficientNetV2-M, EfficientNetV2-B2, EfficientNetV2-B3. Architektury se liší počtem parametrů, počtem konvolučních bloků, dimenzí výstupních příznaků a také na jakém datasetu byly předtrénované – ImageNet-1k [40] nebo ImageNet-21k [41]. Přehled použitých architektur je zobrazen v následující tabulce (Tabulka 4.1), včetně jejich charakteristik a url odkazů na Hugging Face. Počet parametrů modelu neobsahuje parametry klasifikační vrstvy.

[8] https://huggingface.co/timm/tf_efficientnetv2_b2.in1k
[9] https://huggingface.co/timm/tf_efficientnetv2_b3.in1k
[10] https://huggingface.co/timm/tf_efficientnetv2_s.in1k
[11] https://huggingface.co/timm/tf_efficientnetv2_m.in21k
[12] https://huggingface.co/timm/tf_efficientnetv2_m.in1k

Tabulka 4.1 Srovnání architektur EfficientNetV2.

Model	#Parametrů	#Konv. bloků	Dataset	#Příznaků d
EfficientNetV2-B2-in1k[8]	8,7 M	6	ImageNet-1k	1408
EfficientNetV2-B3-in1k[9]	12,8 M	7	ImageNet-1k	1536
EfficientNetV2-S-in1k[10]	20,1 M	6	ImageNet-1k	1280
EfficientNetV2-M-in1k[11]	52,9 M	7	ImageNet-1k	1280
EfficientNetV2-M-in21k[12]	52,9 M	7	ImageNet-21k	1280

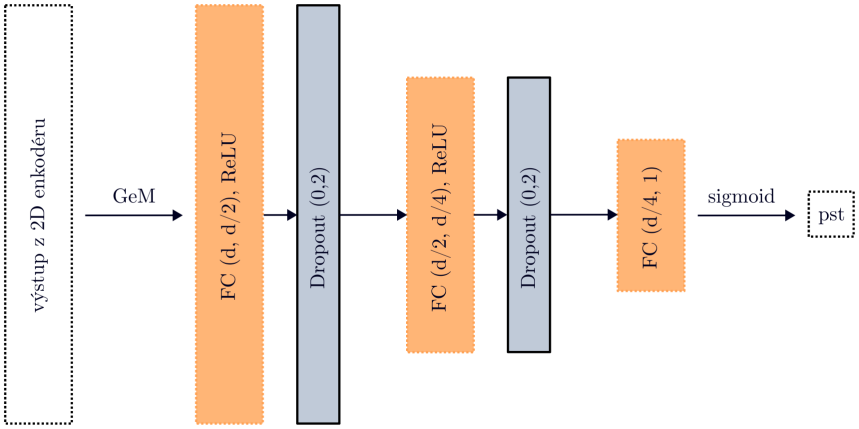
Data vstupující do modelu při tréninku podléhají augmentaci. Augmentace slouží pro zvýšení obecnosti modelu. Data jsou konkrétně transformována operacemi – přeškálování rozlišení, náhodné horizontální převrácení, afinní škálování a rotace, barevná úprava a vymaskování částí obrazu. Afinní škálování zvětší snímek až o 5 % a orotuje o 0 až 10 stupňů, čímž simuluje klepání či rotaci kamery. Barevná transformace upravuje náhodně jas, kontrast, saturaci a barevnost obrazu, kde váhy úprav jsou v poměru 0,6 : 0,2 : 0,3 : 0,2, za účelem adaptace na různé podmínky snímání. Vymaskování částí obrazu generuje maximálně 6 černých čtverců o maximální velikosti 24×24 pixelů, které jsou náhodně rozprostřeny do obrazu a zabraňují modelu predikovat pouze na základně specifické části. Náhodné horizontální převrácení simuluje nastavení kamery, kdy se hlavní rozhodčí nachází na pravé nebo na levé straně kurtu.

Obrázek 4.2 Augmentace vstupních dat – originál bez transformace vlevo nahoře.

Štítky sousedících snímků se snímky výměn jsou upravené metodou label smoothingu. Snímky, které leží do jedné sekundy od začátku nebo konce výměny, mají upravenou hodnotu na 0,25. Snímky ležící do dvou sekund od výměny na obě strany jsou ohodnoceny číslem 0,5. Výsledný

efekt vyhlazuje hrany ohodnocení mezi výměnou a „nevýměnou“. Špatné ohodnocení snímku třídou `1` v blízkosti opravdové výměny je tedy méně penalizováno, čímž je docíleno lehké tolerance při trénování.

Klasifikační hlava, připojená po extrakci příznaků konvoluční sítí (2D enkodéru) a aplikaci GeM poolingu, se skládá ze tří FC vrstev s ReLU aktivací a dvou Dropout vrstev s parametrem 0,2. Parametr p_d pro GeM pooling je nastaven jako $p_d = 3$ uniformně pro všechny kanály výstupu. Počet vstupních a výstupních parametrů závisí na dimenzi výstupních příznaků z enkodéru (Obrázek 4.3). Aktivace sigmoid funkcí je aplikována pouze ve fázi inference, při tréninku aktivaci zajišťuje ztrátová funkce.



Obrázek 4.3 Architektura klasifikační hlavy, d značí počet příznaků výstupu z 2D enkodéru.

Výstupem modelu jsou pravděpodobnosti jednotlivých snímků, zda se jedná o snímky výměny či ne. Pravděpodobnosti jsou zaokrouhleny a tím jsou získány třídy `0/1`. Zaokrouhlené hodnoty jsou vyhlazeny mediánovým filtrem s velikostí kernelu 7×7 za účelem odstranění šumových výměn. Intervaly po sobě jdoucích jedniček reprezentují časové značky jednotlivých výměn. Pouze intervaly trvající více než 3 sekundy představují výsledné časové značky.

Trénování probíhalo na grafické kartě NVIDIA V100 s použitím AdamW optimalizačního algoritmu [42] a plánovače s kosinovým vyhlazováním [43]. S ohledem na problém binární klasifikace (výměna/nevýměna), byl trénink evaluován pomocí ztrátové funkce binární křížové entropie. Konkrétně se jedná o PyTorch implementaci `torch.nn.BCEWithLogitsLoss`[13], která kombinuje binární cross-entropii s aktivační funkcí sigmoid a je tím stabilnější než postupná aplikace sigmoidy a ztrátové funkce.

Následující hyperparametry byly modifikovány během tréninku: počet epoch, learning rate a architektura modelu. Velikost batche pro všechny experimenty byla nastavena na 16 snímků, rozlišení vstupních dat bylo konstantní s rozměry 510×510 pixelů, a snímkovací frekvence odpovídala jednomu snímku za sekundu. Model je trénován jako celek – enkodér i klasifikační hlava, přičemž enkodér má inicializované váhy z tréninku na ImageNet-1k/ImageNet-21k datasetu.

Experimenty probíhaly na sloučeném datasetu skupin 1 a 3 (Tabulky A.1, A.3) čítající 91 trénovacích videí, 22 videí validačních. Architektura vedoucí na nejpřesnější střih byla dále použita při tréninku finálního modelu na sloučeném datasetu všech čtyř skupin (Tabulky A.1,A.2,

[13] https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

A.3 A.4). Finální model je testován na validačním i testovacím datasetu. Doba trvání tréninku se v průměru pohybuje mezi jednou hodinou až hodinou a půl na jednu epochu, dle velikosti modelu. Inference s architekturou EfficientNetV2-B3 trvá v průměru 15 sekund na jeden zápas při paralelním načítání dat na 16 vláknech.

4.3 2.5D klasifikace

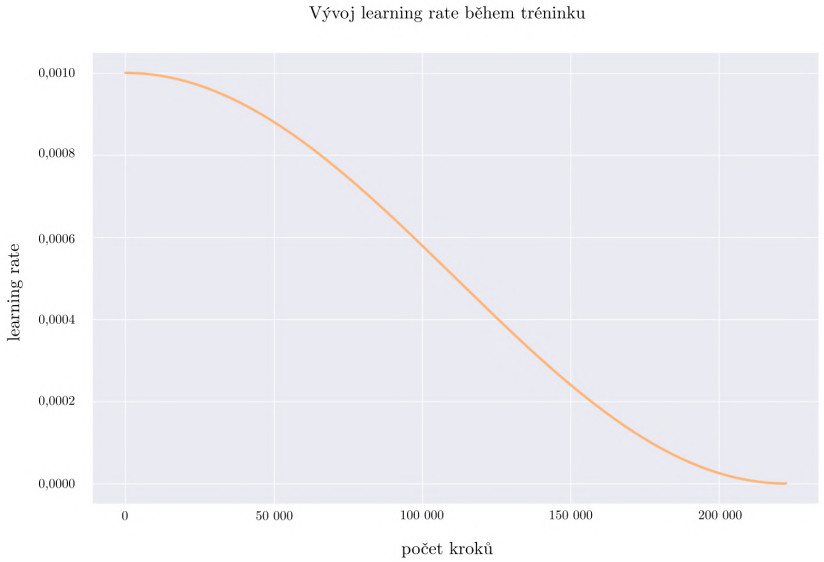
Realizace 2.5D klasifikace vychází z experimentů 2D klasifikace. Modelem pro 2.5D klasifikaci je konvoluční neuronová síť architektury EfficientNetV2-B3, která prokázala nejlepší úspěšnost v předchozí implementaci. Na rozdíl od 2D klasifikace, kdy je klasifikován každý snímek nezávisle na ostatních, vstupuje do modelu shluk N konkatenovaných RGB snímků a klasifikace středového snímku je ovlivněna snímky okolními. Klasifikační hlava kopíruje stejnou architekturu tří FC vrstev (viz. Obrázek 4.3), kde při použití EfficientNetV2-B3 parametr d náleží 1536 příznakům. Konkrétně je aplikován GeM pooling na výstup z 2D enkodéru s parametrem $p_d = 3$ uniformně pro všechny kanály výstupu. První FC vrstva má 1536 vstupních neuronů a 768 neuronů výstupních, druhá FC vrstva obsahuje 768 vstupních a 384 výstupních neuronů. Poslední FC vrstva má k dispozici 384 neuronů a na výstupu pouze 1.

Vstupní data jsou augmentována stejnými transformacemi jako v sekci 4.2: přeškálování rozlišení, náhodné horizontální převrácení, afinní škálování a rotace, barevná úprava a vymaskování částí obrazu. Na snímky v rámci jednoho shluku je aplikována stejná transformace, aby byla zachována podstata podobně vypadajících sousedních snímků videa. Augmentace využívají knihovnu Albumentations [44]. Štítkům snímků sousedících s výměnou je přiřazeno ohodnocení 0,25 nebo 0,5 dle vzdálenosti v sekundách.

Výstupem modelu jsou pravděpodobnosti příslušnosti k třídě `1` středových snímků každého shluku. Počet shluků je roven počtu snímků videa dle snímkovací frekvence. Na rozdíl od implementace 2D klasifikace proběhly experimenty s větší snímkovací frekvencí při inferenci, konkrétně se dvěma snímky za sekundu, kde byl testován scénář větší snímkovací frekvence než při tréninku s následnou možností průměrování výsledných predikcí v rámci jedné sekundy videa. Časová řada zaokrouhlených pravděpodobností je vyhlazena mediánovým filtrem velikosti 9×9. Intervaly po sobě jdoucích jedniček o délce větší než 4 sekundy definují výstup v podobě časových značek jednotlivých výměn videa.

Trénink modelu probíhal na grafické kartě NVIDIA V100 s použitím AdamW optimalizačního algoritmu [42] a plánovače s kosinovým vyhlazováním [43]. Kosinové ochlazování postupně snižuje learning rate s počtem kroků tréninku dle dané křivky (Obrázek 4.4). Díky charakteru binární klasifikace je ztrátovou funkcí binární křížové entropie, konkrétně PyTorch implementace kombinující ztrátovou funkci s aplikací aktivační funkce sigmoid – `torch.nn.BCEWithLogitsLoss`.

Během experimentů byly laděny hyperparametry počtu epoch, learning rate, ale také parametry týkající se vstupních dat – počet konkatenovaných snímků N, velikost kroku k. Rozlišení snímků bylo zafixováno na 512×512 pixelů při snímkovací frekvenci jeden snímek za sekundu. Jeden batch při tréninku obsahuje 16 nebo 32 jednotek konkatenovaných snímků. Experimenty probíhaly všech skupinách datasetu dohromady (Tabulky A.1, A.2, A.3, A.4), ale i na kombinaci podmnožiny – 1 a 3. Dataset obsahující všechny skupiny čítá 221 trénovacích videí, 57 validačních videí a 46 videí testovacích. Trénovací množina 221 videí při 1 fps je složena dohromady z 712 405 snímků. Průměrná doba jedné epochy při použití všech čtyř datasetů dohromady dosahuje ke 4,5 hodinám. Z toho důvodu byly některé experimenty prováděny na menších podmnožinách datasetu. Inference se průměru pohybuje v řádu jednotek minut na jeden zápas dle velikosti shluku a snímkovací frekvence při paralelním načítání na 10 vláknech.



Obrázek 4.4 Kosinové ochlazovaní pro počáteční learning rate 0,001.

Velikost shluku N velmi ovlivňuje dobu tréninku i inference, včetně paměťové náročnosti. Z toho důvodu byla testována velikost shluku 3, 5 a 7 snímků při snímkovací frekvenci jednoho snímku za sekundu. Při velikosti 7 snímků shluk pokrývá přibližně poloviční dobu průměrné délky jedné výměny – 13 sekund. Dále bylo experimentováno s vyšší snímkovací frekvencí při inferenci. Při vyšší snímkovací frekvenci se poměrově zvyšuje i krok, při kterém byl model natrénovaný. Výsledná transformace predikcí klasifikace na časové značky střihu jednotlivých výměn testuje dva přístupy. První průměruje hodnoty predikce stejné sekundy. Naopak druhý přístup ponechává vyšší snímkovací frekvenci a výsledné časové značky jsou na závěr poděleny snímkovací frekvencí, aby byly ponechány jednotky sekund časových značek. Testována je také kombinace (ensembling) dvou modelů dohromady s cílem zlepšení efektivity či většího zorného pole algoritmu.

Kapitola 5
Výsledky

Výsledky lze rozdělit dle přístupu na část optical flow, 2D klasifikace a 2.5D klasifikace. Nastavení hyperparametrů algoritmů bylo testováno na validační množině datasetů a poté následně ohodnoceno na množině testovací. Součástí testovací sady jsou i dva nové datasety z aktuální beachvolejbalové sezóny 2024.

5.1 Optical Flow

Experimenty probíhaly na validační části datasetu č. 2 o velikosti 17 videozáznamů. Testovány byly hyperparametry nastavení algoritmu a zpracování časové řady:

- Rozlišení vstupního videa – původní rozlišení, poloviční rozlišení a čtvrtinové rozlišení
- Snímkovací frekvence – 1, 2 nebo 5 snímků za sekundu
- Velikost kernelu při vyhlazení mediánovým filtrem (MED) nebo klouzavým průměrem (MA)
- Amplituda při rozkladu časové řady (ROZ) dle aditivního modelu – amplituda 10-14 sekund

Kromě hyperparametrů byl testován i způsob vyhlazení časové řady – v jakém pořadí a kombinaci jsou operace prováděny. Volba parametrů a pořadí operací jsou zpracovány vyhledáváním v mřížce. Nejlepší výsledky při zafixovaném polovičním rozlišení jsou zobrazeny v následujících tabulkách (Tabulky 5.1, 5.2, 5.3) rozdělených podle snímkovací frekvence. Experimenty byly porovnávány dle metriky IoU.

Tabulka 5.1 Výsledky optical flow při snímkovací frekvenci 1 fps.

Exp	Způsob	MED	MA	ROZ	IoU	FP	FN	Err start	Err end
1	MA	-	3	-	0,335	47,4	17,8	4,1	5,0
2	ME	5	-	-	0,340	44,4	20,3	4,1	4,7
3	ME-MA	9	3	-	0,340	48,3	12,0	4,2	4,9
4	ROZ	-	-	12	0,371	37,9	13,0	5,1	4,7
5	**ME-ROZ**	**11**	**-**	**12**	**0,383**	**32,7**	**15,8**	**5,7**	**5,0**
6	MA-ME-ROZ	7	3	11	0,366	38,9	12,9	5,2	4,8

Tabulka 5.2 Výsledky optical flow při snímkovací frekvenci 2 fps.

Exp	Způsob	MED	MA	ROZ	IoU	FP	FN	Err start	Err end
1	MA	-	3	-	0,287	49,1	25,1	3,8	5,5
2	ME	17	-	-	0,307	51,0	13,9	4,1	4,9
3	MA-ME	17	3	-	0,291	50,2	19,1	4,0	5,1
4	ME-MA	17	3	-	0,318	51,5	10,9	4,0	5,0
5	ROZ	-	-	13	0,372	37,3	12,3	5,0	4,8
6	**ME-ROZ**	**15**	**-**	**12**	**0,385**	**39,2**	**9,5**	**4,7**	**4,7**

Tabulka 5.3 Výsledky optical flow při snímkovací frekvenci 5 fps.

Exp	Způsob	MED	MA	ROZ	IoU	FP	FN	Err start	Err end
1	MA	-	3	-	0,208	44,1	42,2	3,9	6,5
2	ME	41	-	-	0,262	52,5	20,4	4,0	5,7
3	MA-ME	31	3	-	0,224	50,1	34,1	4,1	6,1
4	ME-MA	31	3	-	0,245	51,4	23,8	3,8	6,2
5	ROZ	-	-	13	0,365	35,9	13,6	5,4	5,0
6	**ME-ROZ**	**41**	**-**	**13**	**0,372**	**36,1**	**11,8**	**5,4**	**5,0**

U všech tří snímkovacích frekvencí nejlepšího IoU skóre dosáhla aplikace mediánu a rozkladu časové řady dle aditivního modelu s rozdílem ve velikosti aplikovaného filtru a amplitudy při rozkladu. Velikost mediánového filtru se přirozeně zvětšuje se snímkovací frekvencí, zatímco amplituda vyjádřena v sekundách zůstává na hodnotě 12/13 sekund.

Nejvyšší hodnoty IoU skóre – 0,385 – dosáhla konfigurace při snímkovací frekvenci dvou snímků za sekundu spolu s vyhlazením mediánovým filtrem o rozměru 15×1 a amplitudou 12 sekund. Zároveň průměrný počet extra vygenerovaných výměn (FP) je 39,2 a naopak v průměru 9,5 výměn nebylo zdetekováno (FN). Chyba začátku výměny je v průměru 4,7 sekundy a chyba konce výměny také 4,7 sekundy.

Nejúspěšnější konfigurace každé testované snímkovací frekvence je výchozí konfigurací pro experiment ověření závislosti rozlišení na úspěšnosti. Výsledky jsou zobrazeny v následující tabulce (Tabulka 5.4).

Z výsledků není patrná závislost rozlišení na úspěšnosti. V případě snímkovací frekvence 1 fps kleslo IoU skóre při zvýšení i snížení rozlišení. Experiment č. 8 při 5 fps ukazuje lehké zlepšení oproti polovičnímu rozlišení. Experiment č. 4 je experimentem dosahujícím nejlepšího IoU skóre ze všech experimentů, kdy zvýšení rozlišení na původní hodnotu zvýšilo průměrné IoU skóre o 3 % oproti rozlišení 960×540 pixelů.

Nejlepšího IoU skóre (0,590) dosahuje video z turnaje Challenger Edmont na kurtu C3, kde se chybovost začátku/výměny pohybuje okolo 2 sekund. Pouze 3 skutečné výměny nebyly zdetekovány, naopak počet navíc vygenerovaných výměn dosáhl hodnoty 19, ale jedná se o výměny před začátkem zápasu, po konci zápasu a mezi pauzami. Zbytek vygenerovaných výměn odpovídá skutečnosti. Na následujícím obrázku (Obrázek 5.1a) je zobrazen pohled z kamery a graf predikce výměn (Obrázek 5.1b).

Z výsledků je patrné, že algoritmus není dostatečně silný pro řešení automatického střihu videa. Výsledky trpí vysokým počtem extra vygenerovaných výměn a zároveň i velkým počtem skutečných výměn, které nebyly rozpoznány. Optical flow nerozpoznává, zda měřený pohyb je opravdu vykonáván hráči, nebo například zametači kurtů či roztleskávačkami během pauz.

Tabulka 5.4 Výsledky optical flow při variabilním rozlišení.

Exp	fps	Rozlišení	IoU	FP	FN	Err start	Err end
1	1	1920 × 1080	0,356	31,8	19,1	6,4	5,4
2	1	960 × 540	0,383	32,7	15,8	5,7	5,0
3	1	480 × 270	0,381	33,6	15,4	5,6	4,9
4	**2**	**1920 × 1080**	**0,397**	**36,1**	**11,3**	**4,7**	**4,5**
5	2	960 × 540	0,385	39,2	9,5	4,7	4,7
6	2	480 × 270	0,387	38,8	9,9	4,7	4,6
7	5	960 × 540	0,372	36,1	11,8	5,4	5,0
8	5	480 × 270	0,375	35,8	11,3	5,3	5,0

5.2 2D klasifikace

Experimenty probíhaly na spojeném datasetu č. 1 a č. 3 o velikosti 91 tréninkových videí, 22 videí validačních a 19 testovacích videí. Testovány byly hyperparametry pro trénink modelu:

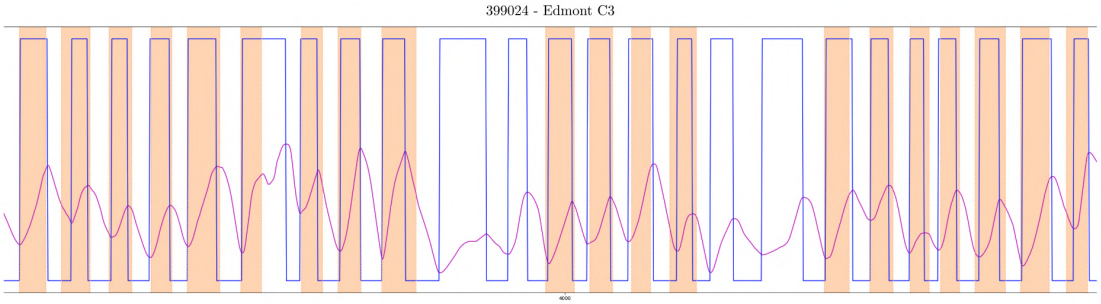
- Architektura 2D enkodéru – typ EfficientNetV2
- Počet epoch tréninku
- Počáteční learning rate

Snímkovací frekvence byla zafixována na 1 fps, stejně tak rozlišení na 512 × 512 pixelů a velikost batche na 16 snímků. Experimenty cílily na porovnání použitých architektur při extrakci příznaků ze snímků. Následující tabulka obsahuje nastavení hyperparametrů dosahujících nejlepších výsledků pro pět testovaných architektur (Tabulka 5.5). Výsledky jsou evaluovány dle IoU skóre, přesnosti klasifikace (Acc), počtu extra vygenerovaných výměn (FP) a počtu nezdetekovaných skutečných výměn (FN). Vývoj metrik při tréninku je zobrazen na následujícím obrázku (Obrázek 5.2), kde je viditelné zlepšení každé metriky s rostoucí epochou.

Tabulka 5.5 Srovnání výsledků dle architektur EfficientNetV2.

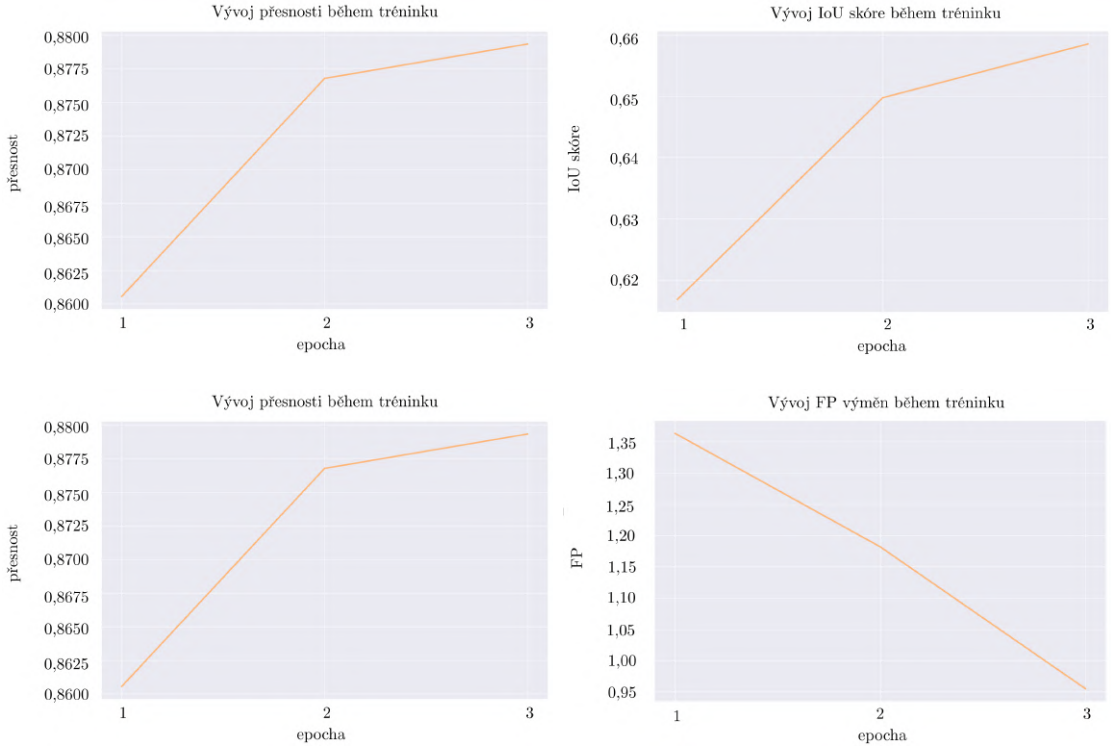
Model	#Epoch	Learning rate	Acc	IoU	FP	FN
EfficientNetV2-B2-in1k	3	$1e-4$	0,866	0,630	0,59	10,50
EfficientNetV2-B3-in1k	**3**	**1e − 3**	**0,879**	**0,658**	**0,95**	**5,14**
EfficientNetV2-S-in1k	4	$8e-4$	0,873	0,638	0,82	6,32
EfficientNetV2-M-in1	6	$8e-4$	0,859	0,624	1,14	10,32
EfficientNetV2-M-in21k	5	$1e-3$	0,848	0,607	0,77	16,0

(a) Pohled z kamery.



(b) Část grafu predikce výměn.

Obrázek 5.1 Challenger Edmont C3, zápas 399024.



Obrázek 5.2 Vývoj metrik při tréninku pro architekturu EfficientNetV2-B3-in1k.

Nejlépe vyhodnocená architektura – EfficientNetV2-B3-in1k – byla použita při tréninku na datasetu všech skupin dohromady (dataset 1234). Model byl trénován 4 epochy s learning ratem $1e{-}3$ v celkovém čase 8 hodin. V následujících dvou tabulkách je zobrazena úspěšnost algoritmu na validační a testovací části datasetu 1234. První tabulka obsahuje metriky zaměřující se na klasifikaci a jednotlivé výměny (Tabulka 5.6). Druhá tabulka se zaměřuje na metriky správnosti střihu (Tabulka 5.7).

Tabulka 5.6 Úspěšnosti 2D klasifikace na datasetu 1234 – klasifikace.

Data	Acc	F1	IoU	FP	FN	Multi GTs	Multi Preds	Err start	Err end
val	0,897	0,858	0,710	0,60	1,63	0,14	0,14	2,43	1,77
test	0,905	0,865	0,720	0,49	2,43	0,00	0,11	2,17	1,84

Tabulka 5.7 Úspěšnosti 2D klasifikace na datasetu 1234 – střih.

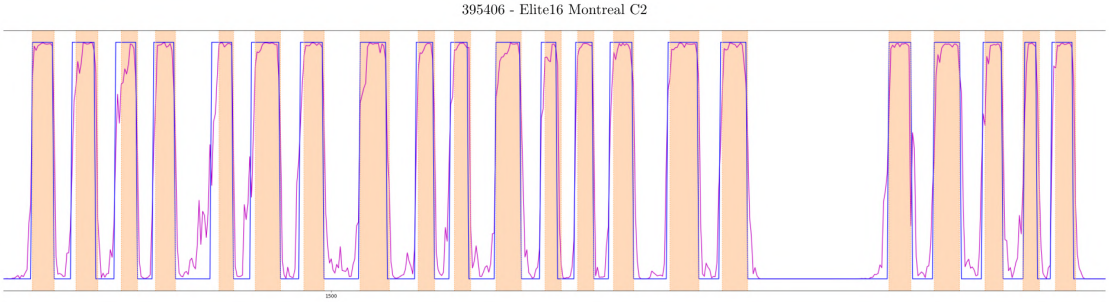
Data	Velikost	Správně	1 chyba	2 chyby	Precision	Recall	FDR
val	57	24 (42 %)	11 (19 %)	6 (11 %)	0,993	0,980	0,007
test	45	21 (47 %)	8 (18 %)	4 (9 %)	0,994	0,973	0,006

Z hlediska správnosti střihu model přesně rozstříhá 42 % zápasů z validační množiny. Pokud jsou tolerovány maximálně 2 chybné výměny, algoritmus dosáhne 71% úspěšnosti v rámci této tolerance. Precision vyjadřující pravděpodobnost, že predikovaná výměna je opravdovou výměnou, dosahuje v průměru 99,3 %. Recall, pravděpodobnost, že existující výměna bude detekována, dosahuje v průměru 98,0 %. Situace predikované výměny, která je výměnou navíc, nastává v 0,7 % výměn (FDR). Na testovací množině precision dosahuje vyšší hodnoty 99,4 %, naopak recall – poměr počtu zdetekovaných výměn ku počtu všech skutečných výměn – klesl na 97,3 %. Při toleranci 2 chybných výměn algoritmus správně nastříhá 73 % zápasů, bez chyby 47 % zápasů. Z tabulky a výsledků je patrné, že model vykazuje podobné výsledky jak na validační, tak na testovací množině.

Nejlepšího výsledku z validační množiny dosahuje video zápasu (id zápasu: 395406) z turnaje z Elite16 Montreal na kurtu C2. Predikovaný střih videa má metriku IoU skóre rovnou 0,850 a všechny metriky střihu (FP, FN, Multi Preds/GTs) jsou rovny nule. Chybovost začátku výměny je v průměru 1,28 sekundy, konce výměn jsou zatížené průměrnou chybu pouhých 0,63 sekund. Naopak nejnižším IoU skóre 0,467 bylo ohodnoceno video zápasu (id zápasu: 378726) z turnaje Elite16 Uberlandia na kurtu C2. Algoritmus nezdetekoval 27 skutečných výměn a zároveň vygeneroval 1 výměnu navíc mimo skutečné výměny. Turnaj Elite16 v Uberlandii patří do datasetu skupiny č. 1 a žádné video téhož turnaje není součástí trénovací množiny, pouze validační a testovací. Právě tyto videa testují schopnost modelu generalizovat na zcela nových datech. Na následujících obrázcích jsou zobrazeny pohledy z kamery a grafy predikce výměn pro nejlepší video z validační množiny (Obrázek 5.3) a pro nejhůře hodnocené video (Obrázek 5.4).

Ve vizualizaci výsledného střihu zápasu v Uberlandii (Obrázek 5.4b) je patrné, že model 2D klasifikace správně identifikuje oblasti, kde by se výměna měla nacházet, avšak predikovaná pravděpodobnost je nedostatečná a krátkodobá. Při převodu na časové značky algoritmus interpretuje signál jako nedostatečně výrazný a spojitý, což vede k ohodnocení snímků třídou `0` – neobsahující výměnu.

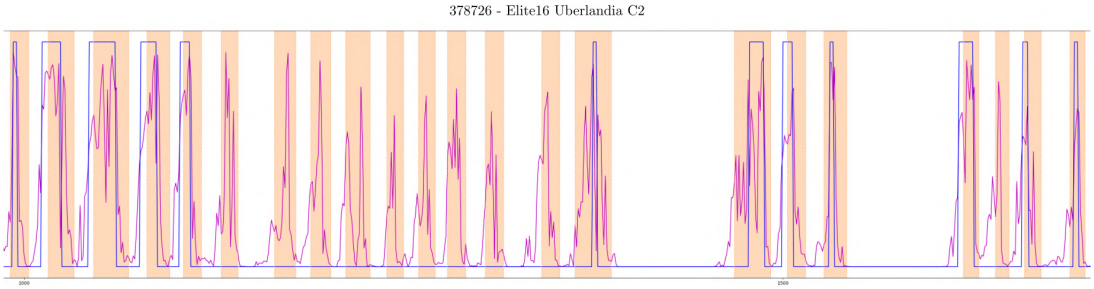
(a) Pohled z kamery.



(b) Část grafu predikce výměn.

Obrázek 5.3 Elite16 Montreal C2, zápas 395406.

(a) Pohled z kamery.



(b) Část grafu predikce výměn.

Obrázek 5.4 Elite16 Uberlandia C2, zápas 378726.

5.3 2.5D klasifikace

Experimenty probíhaly na dvou datasetech: spojeném datasetu skupin č. 1, č. 3 (dataset 13) a datasetu všech skupin dohromady (dataset 1234). Oproti 2D klasifikaci přibyly k hyperparametrům modelu hyperparametry vstupních dat:

- Počet snímků ve shluku (N)
- Rozestupy sousedících snímků – krok (k)

5.3.1 Hyperparametry vstupních dat

Úspěšnost automatického střihu byla evaluována dle metrik popsaných v sekci 3.2.4 na validační části datasetu 1234. Primární metrikou pro seřazení jednotlivých experimentů dle úspěšnosti je metrika IoU skóre, v případě podobného skóre je sekundární porovnání na základě úspěšnosti střihu. V následující tabulce (Tabulka 5.8) jsou také zobrazeny chyba začátku/konce výměny v sekun-dách, přesnost klasifikace (Acc) a F1 skóre. Experimenty byly zaměřeny na hyperparametry vstupních dat – počet snímků ve shluku a krok. Learning rate startoval na hodnotě $1e-3$ a pomocí kosinového ochlazování postupně klesal (Obrázek 4.4).

Tabulka 5.8 Srovnání výsledků dle hyperparametrů vstupních dat.

Exp	#Epoch	N	k	Acc	F1	IoU	FP	FN	Err start	Err end
1	4	3	1	0,914	0,882	0,749	0,44	1,05	2,13	1,58
2	4	3	3	0,917	0,885	0,751	0,42	0,68	2,07	1,59
3	**5**	**5**	**1**	**0,920**	**0,891**	**0,758**	**0,45**	**0,25**	**2,04**	**1,47**
4	**5**	**5**	**2**	**0,917**	**0,887**	**0,756**	**0,47**	**1,42**	**2,07**	**1,50**
5	**5**	**5**	**3**	**0,921**	**0,893**	**0,763**	**0,44**	**0,51**	**2,00**	**1,49**
6	**6**	**7**	**1**	**0,919**	**0,891**	**0,755**	**0,58**	**0,16**	**2,10**	**1,49**
7	5	7	2	0,915	0,883	0,752	0,49	1,77	2,02	1,52

Experimenty č. 3, 4, 5, 6 dosahují nejvyšších IoU skóre. Pro úplnost a porovnání obsahuje následující tabulka (Tabulka 5.9) metriky správnosti střihu. Experimenty č. 3 a 5 založeny na velikosti shluku o 5 snímcích a rozdílných krocích patří mezi nejúspěšnější konfigurace na základě IoU skóre, F1 skóre i přesnosti klasifikace. Experiment č. 3 je vyhodnocen jako nejlepší, i když IoU skóre dosahuje nižší průměrné hodnoty než konfigurace při experimentu č. 5. Nicméně úspěšnost správnosti střihu při toleranci do 2 chyb činí 96 %, v druhém případě je do dvou chyb nastříháno pouze 87 % zápasů.

Tabulka 5.9 Úspěšnost 2.5D klasifikace – střih.

Exp	Velikost	Správně	1 chyba	2 chyby	Precision	Recall	FDR
3	**57**	**33 (58 %)**	**13 (23 %)**	**9 (16 %)**	**0,995**	**0,997**	**0,005**
4	57	31 (54 %)	12 (21 %)	3 (5 %)	0,996	0,977	0,004
5	57	29 (51 %)	15 (26 %)	6 (11 %)	0,995	0,994	0,005
6	57	36 (63 %)	9 (16 %)	2 (4 %)	0,996	0,993	0,004

Rozlišení vstupních snímků bylo po dobu experimentů zafixováno na 512×512 pixelů. Výjimkou je experiment závislosti vstupního rozlišení na výstupní úspěšnosti probíhající na datasetu 13. Byly sjednoceny hyperparametry tréninku a vstupních dat: 3 epochy, $1e-3$ learning rate, 5 snímků ve shluku a krok k roven jedné. Výsledky pro 3 různá rozlišení jsou v následující tabulce (Tabulka 5.10). Při zvyšování rozlišení lehce stoupá přesnost klasifikace a F1 skóre. Největší skok mezi nejnižším a nejvyšším rozlišení experimentu nastává v metrice false negative snímků (FN), která klesla z průměrných 6,27 snímků na 0,55 snímků.

Tabulka 5.10 Porovnání výsledků 2.5D klasifikace při variabilním rozlišení.

Rozlišení	Acc	F1	IoU	FP	FN	Err start	Err end
384×384	0,893	0,843	0,710	0,91	6,27	2,32	1,88
512×512	0,903	0,861	0,719	0,50	2,14	2,26	1,84
810×810	0,907	0,866	0,715	0,82	0,55	2,22	1,90

5.3.2 Snímkovací frekvence

Modely byly trénovány na trénovací množině datasetu 1234 se snímkovací frekvencí jednoho snímku za sekundu. Zdvojnásobení snímkovací frekvence při inferenci bylo testováno na dvou modelech z experimentu č. 3 a č. 5 – konfigurace při tréninku odpovídají 5 snímkům shluku a kroku $k = 1$ a $k = 3$. Při zdvojnásobení snímkovací frekvence se zdvojnásobuje krok k, aby byl zachován stejný rozestup v sekundách jako při tréninku modelu. Zpracování výstupu modelu v podobě pravděpodobností, zda každý středový snímek shluku je snímkem výměny, probíhalo 2 způsoby:

- Průměrováním predikcí (avg) – predikce v každé sekundě videa je zprůměrována, čímž se transformuje výsledná časová řada predikcí do frekvence jednoho snímku za sekundu a je možné vytvořit intervaly časových značek pro jednotlivé výměny.
- Konverzí finálních časových značek – časová řada je zpracována ve vyšší frekvenci a finální časové značky jsou konvertovány do jednotek sekund.

V následujících tabulkách (Tabulka 5.11, 5.12) jsou zobrazeny výsledky obou způsobu zpracování, včetně výsledků původní konfigurace snímkovací frekvence 1 fps. Časová řada zaokrouhlených pravděpodobností je vyhlazena mediánovým filtrem o zvětšené velikosti (15×15) oproti časové řadě s frekvencí jednoho snímku za sekundu. Inference 57 validačních videí při načítání dat na 10 vláknech trvá 1 hodinu a 51 minut, což v průměru odpovídá 1,9 minuty na video. Průměrná doba strávená při predikci jednoho videa je dvojnásobná oproti poloviční snímkovací frekvenci.

Experimenty ukázaly, že zvýšením snímkovací frekvence se převážně sníží průměrná absolutní chyba konce výměny a zvýší IoU skóre. Při aplikaci průměrování došlo sice ke zvýšení přesnosti klasifikace a F1 skóre, ale výrazně pokleslo IoU skóre. Z pohledu úspěšnosti střihu došlo spíše ke zhoršení. FDR se zvýšilo u obou způsobů zpracování, naopak precision výměn kleslo.

5.3.3 Ensembling

Kombinace více modelů řešících stejný problém může vést ke zlepšení stability celkového modelu [45]. Za tímto účelem byl testován ensembling modelů z experimentu 3 a 5, které přijímají shluk o velikosti 5 snímků s rozdílným krokem k a snímkovací frekvencí **jednoho snímku za sekundu**. Při kombinaci rozdílných kroků k se rozšiřuje zorné pole modelu. Na následujícím

Tabulka 5.11 Výsledky zpracování vyšší snímkovací frekvence – model z experimentu č. 3 a 5.

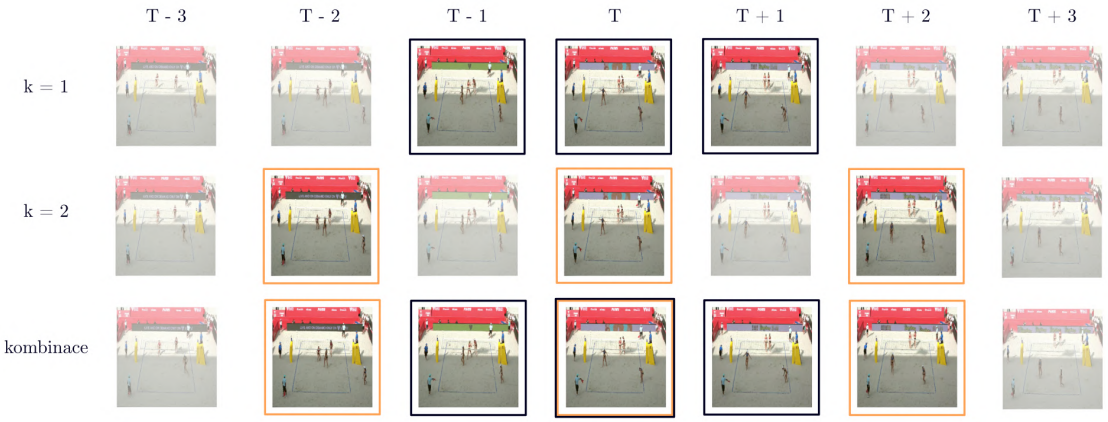
Exp	fps	Acc	F1	IoU	FP	FN	Err start	Err end
3	1	0,920	0,891	0,758	0,45	0,25	2,04	1,47
3	**2**	**0,917**	**0,888**	**0,763**	**0,47**	**0,19**	**2,14**	**1,32**
3	avg	0,921	0,892	0,739	0,47	0,26	2,13	1,69
5	1	0,921	0,893	0,763	0,44	0,51	2,00	1,49
5	**2**	**0,918**	**0,889**	**0,767**	**0,51**	**0,37**	**2,11**	**1,34**
5	avg	0,921	0,893	0,743	0,47	0,45	2,10	1,69

Tabulka 5.12 Výsledky střihu při vyšší snímkovací frekvenci – model z experimentu č. 3 a 5.

Exp	fps	Správně	1 chyba	2 chyby	Precision	Recall	FDR
3	1	33 (58 %)	13 (23 %)	9(16 %)	0,995	0,997	0,005
3	2	35 (61 %)	12 (21 %)	7 (12 %)	0,992	0,998	0,008
3	avg	32 (56 %)	11 (19 %)	5 (9 %)	0,994	0,997	0,006
5	1	29 (51 %)	15 (26 %)	6 (11 %)	0,995	0,994	0,005
5	2	30 (53 %)	16 (28 %)	4 (7 %)	0,993	0,993	0,007
5	avg	30 (53 %)	10 (18 %)	5 (9 %)	0,994	0,994	0,006

obrázku je vymodelovaná situace rozšíření zorného pole pro shluk 3 snímků a krokem $k = 1, k = 2$ (Obrázek 5.5). Při shluku velikosti 5 snímků a kroku $k = 1, k = 3$, je predikce pro snímek T ovlivněna snímky S dle rovnice

$$S = \{\, T + x \mid x \in [-6, -3, -2, -1, 0, 1, 2, 3, 6] \,\}. \tag{5.1}$$



Obrázek 5.5 Rozšíření zorného pole modelu pro shluk o velikosti 3 snímků.

Při kombinaci dvou modelů byly testovány váhy, jak moc jednotlivé modely přispívají k výsledku. Nejjednodušší kombinací je zprůměrování výstupů z obou modelů, tudíž váhy výsledku jsou rovnoměrně rozděleny v poměru 1 : 1. Dále bylo testováno nerovnoměrné rozdělení vah – (0,75, 0,25), (0,6, 0,4), kdy jeden model přispívá k výsledku více než model druhý. V následující

tabulce jsou zobrazeny výsledky pro různé váhy včetně původních výsledků experimentu č. 3 a č. 5 (Tabulka 5.13, 5.14) – spouštěno na validační množině. Z výsledků je zřejmé, že kombinace modelů vždy nepatrně zlepšila celkové IoU skóre i všechny ostatní metriky kromě FN, které kolísá, oproti hodnotám samostatných modelů. Skoro shodných výsledků dosahuje kombinace modelů s váhami (0,5, 0,5), (0,4, 0,6) a (0,6, 0,4), avšak z pohledu úspěšnosti střihu kombinace (0,6, 0,4) nastříhá 98 % videí s tolerancí 2 chyb a 60 % videí bez žádné chyby. Precision – pravděpodobnost, že predikovaná výměna, je skutečnou výměnou – se vyšplhala na 99,6 %.

Tabulka 5.13 Srovnání výsledků ensemblingu dle vah modelů z experimentů č. 3 a 5.

Váha (3)	Váha (5)	Acc	F1	IoU	FP	FN	Err start	Err end
1	0	0,920	0,891	0,758	0,45	0,25	2,04	1,47
0	1	0,921	0,893	0,763	0,44	0,51	2,00	1,49
0,50	**0,50**	**0,925**	**0,899**	**0,767**	**0,42**	**0,28**	**1,95**	**1,43**
0,60	**0,40**	**0,925**	**0,898**	**0,767**	**0,30**	**0,18**	**1,94**	**1,43**
0,75	0,25	0,923	0,895	0,763	0,42	0,23	1,98	1,45
0,40	**0,60**	**0,925**	**0,898**	**0,767**	**0,42**	**0,33**	**1,95**	**1,43**
0,25	0,75	0,924	0,896	0,766	0,40	0,33	1,95	1,44

Tabulka 5.14 Srovnání úspěšnosti střihu při ensemblingu dle vah modelů z experimentů č. 3 a 5.

Váha (3)	Váha (5)	Správně	1 chyba	2 chyby	Precision	Recall	FDR
1	0	33 (58 %)	13 (23 %)	9 (16 %)	0,995	0,997	0,005
0	1	29 (51 %)	15 (29 %)	6 (11 %)	0,995	0,994	0,005
0,50	0,50	33 (58 %)	12 (21 %)	10 (18 %)	0,995	0,996	0,005
0,60	**0,40**	**34 (60 %)**	**15 (26 %)**	**7 (12 %)**	**0,996**	**0,997**	**0,004**
0,75	0,25	34 (60 %)	12 (21 %)	9 (16 %)	0,995	0,997	0,005
0,40	0,60	33 (58 %)	13 (23 %)	7 (12 %)	0,995	0,996	0,005
0,25	0,75	34 (60 %)	11 (19 %)	7 (12 %)	0,995	0,995	0,005

Konfigurace vah (0,6, 0,4) vyhodnocená jako nejúspěšnější pro snímkovací frekvenci jednoho snímku za sekundu, byla aplikována pro vstupní data se snímkovací frekvencí **dvou snímků za sekundu**. Dle výsledků z následujících tabulek (Tabulka 5.15, 5.16) je patrné, že metriky jsou téměř totožné. Při dvojnásobné snímkovací frekvenci bylo zvýšeno IoU skóre o 0,3 %, snížena průměrná absolutní chyba konce výměny o 0,14 sekundy. Naopak průměrná absolutní chyba začátku výměn a průměrný počet skutečných výměn, ke kterým nebyla spárována žádná vypredikovaná výměna (FN), se zvýšily. Z pohledu úspěšnosti střihu algoritmus při snímkovací frekvenci 1 fps nastříhá správně s tolerancí 2 chyb 98 % zápasů, zatímco při dvojnásobné snímkovací frekvenci je v toleranci 2 chyb 93 % zápasů.

Snížení celkové úspěšnosti střihu a delší doba inference při snímkovací frekvenci $fps = 2$ je důvodem ohodnocení modelu s nižší snímkovací frekvencí a následující konfigurací jako nejefektivnější pro automatický střih beachvolejbalových zápasů:

- Snímkovací frekvence – jeden snímek za sekundu
- 1. model – velikost shluku $N = 5$, krok $k = 1$
- 2. model – velikost shluku $N = 5$, krok $k = 3$
- Ensemble 1. a 2. modelu v poměru 0,6 : 0,4

Tabulka 5.15 Srovnání výsledků ensemblingu dle fps vstupních dat.

fps	Váha (3)	Váha (5)	Acc	F1	IoU	FP	FN	Err start	Err end
1	**0,60**	**0,40**	**0,925**	**0,898**	**0,767**	**0,30**	**0,18**	**1,94**	**1,43**
2	0,60	0,40	0,922	0,894	0,770	0,44	0,19	2,07	1,29

Tabulka 5.16 Srovnání úspěšnosti střihu při ensemblingu dle fps vstupních dat.

fps	Váha (3)	Váha (5)	Správně	1 chyba	2 chyby	Precision	Recall	FDR
1	**0,60**	**0,40**	**34 (60 %)**	**15 (26 %)**	**7 (12 %)**	**0,996**	**0,997**	**0,004**
2	0,60	0,40	36 (63 %)	12 (21 %)	5 (9 %)	0,995	0,997	0,005

5.4 Testovací množina

Nejlépe vyhodnocený model dle konfigurace v sekci 5.3.3 byl evaluován na testovací části datasetu 1234, který obsahuje 45 záznamů beachvolejbalových zápasů. Doba běhu inference trvala dohromady pro oba modely 90 minut při paralelním načítání na 10 vláknech, tzn. v průměru 2 minuty na jedno video. Výsledky jsou rozděleny do dvou tabulek (Tabulka 5.19, 5.20). První tabulka je zaměřena na výsledky klasifikace, IoU skóre a metriky jednotlivých výměn. Druhá tabulka nese informaci o úspěšnosti střihu jako celku, včetně metrik precision, recall a FDR.

Tabulka 5.17 Výsledky na testovací množině.

Data	Acc	F1	IoU	FP	FN	Multi GTs	Multi Preds	Err start	Err end
val	0,925	0,898	0,767	0,30	0,18	0.11	0,02	1,94	1,43
test	0,930	0,902	0,772	0,27	0,29	0,02	0,07	1,84	1,44

Tabulka 5.18 Výsledky na testovací množině – střih.

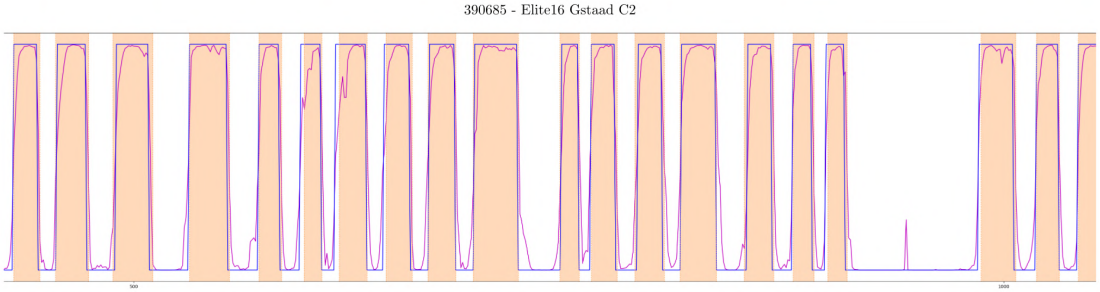
Data	Velikost	Správně	1 chyba	2 chyby	Precision	Recall	FDR
val	57	34 (60 %)	15 (26 %)	7 (12 %)	0,996	0,997	0,004
test	45	29 (64 %)	11 (34 %)	4 (9 %)	0,996	0,997	0,004

Algoritmus drží stejné kvality na validační i testovací sadě. Průměrné IoU skóre na testovací množině dosáhlo hodnoty 77,2 % a přesnost klasifikace 93,0 %. Průměrná absolutní chyba začátku/konce výměny klesla pod 2 sekundy. Algoritmus rozstříhal 64 % zápasů bez chyby. V případě tolerance do 2 chyb algoritmus správně rozstříhá 98 % zápasů v dané toleranci.

Nejvyššího IoU skóre (0,854) dosáhlo video z turnaje Elite16 ve městě Gstaad, na kurtu C2 (id: 390685). Přesnost klasifikace odpovídá 94,1 %, což není maximum napříč datasetem (96,4 %). Zápas byl rozstříhán bez chyb s průměrnou absolutní chybou 1,21 sekundy začátku výměn a 1,04 sekundy konce výměn. Naopak nejnižším IoU skórem (0,598) bylo ohodnoceno video z turnaje Elite16 v německém Hamburgu, na kurtu CC (id: 400998), přesností 90,0 % a F1 skórem 83,8 %, což je minimum napříč datasetem. Zápas byl sice rozstříhán bez chyb, ale průměrná absolutní chyba začátku výměny se vyšplhala na 3,92 sekundy a konec zápasu na 2,46 sekundy. Na následujících obrázcích jsou zobrazeny pohledy z kamery a grafy predikce výměn pro nejlepší video z testovací množiny (Obrázek 5.6) a pro nejhůře hodnocené video (Obrázek 5.7).

Situace, kdy není zdetekována skutečná výměna (FN), nastává při krátce trvající výměně, typicky se jedná o přímý bod z podání či zkažení servisu. Délka výměny toho typu se pohybuje v rozmezí od 5 do 10 sekund. Extra vygenerování výměny (FP) může být např. způsobeno, když rozhodčí odpíská nový míč. Při novém míči se skóre nemění a poslední výměna je opakována. Původní výměna, která byla přerušena, není obsažena ve skutečných datech. V testovací množině nastal také případ, kdy byla vygenerována extra výměna při taneční sestavě roztleskávaček během pauzy mezi sety – pravděpodobně kvůli podobnému úboru a blízkému postavení ke kameře (Obrázek 5.8).

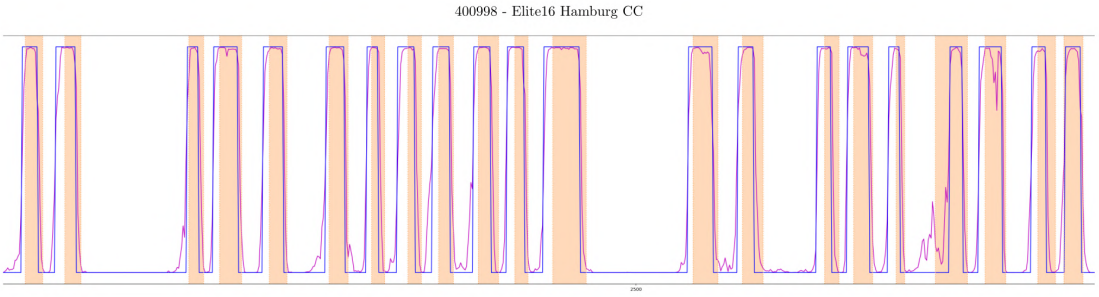
(a) Pohled z kamery.



(b) Část grafu predikce výměn.

Obrázek 5.6 Elite16 Gstaad C2, zápas 390685.

(a) Pohled z kamery.



(b) Část grafu predikce výměn.

Obrázek 5.7 Elite16 Hamburg CC, zápas 400998.

Obrázek 5.8 Vystoupení roztleskávaček během pauzy mezi sety.

5.5 Experiment nového turnaje

Reálnou situaci z praxe simuluje experiment nového turnaje, který prověřuje schopnost modelu dle konfigurace v sekci 5.5 generalizovat na nových datech nové sezóny. Byly vytvořeny dva datasety z počátečních turnajů sezóny 2024 – Challenger v brazilském Recife a Elite16 v katarském Doha. Datasety obsahují pouze videa z hlavní soutěže a čítají 37 a 51 videí. Výsledky jsou rozděleny do dvou tabulek (Tabulka 5.19, 5.20). První tabulka je zaměřena na výsledky klasifikace, IoU skóre a metriky jednotlivých výměn. Druhá tabulka nese informaci o úspěšnosti střihu jako celku, včetně metrik precision, recall a FDR.

Tabulka 5.19 Výsledky na testovací množině.

Data	Acc	F1	IoU	FP	FN	M. GTs	M. Preds	Err start	Err end
Doha 2024	0,922	0,899	0,767	0,33	0,10	0,04	0,02	1,87	1,58
Recife	0,908	0,876	0,736	0,24	0,57	0,05	0,3	2,09	1,79

Tabulka 5.20 Výsledky na testovací množině – střih.

Data	Velikost	Správně	1 chyba	2 chyby	Precision	Recall	FDR
Doha 2024	51	37 (73 %)	6 (12%)	8 (16 %)	0,996	0,994	0,004
Recife	37	23 (62 %)	9 (24 %)	1 (3 %)	0,997	0,994	0,003

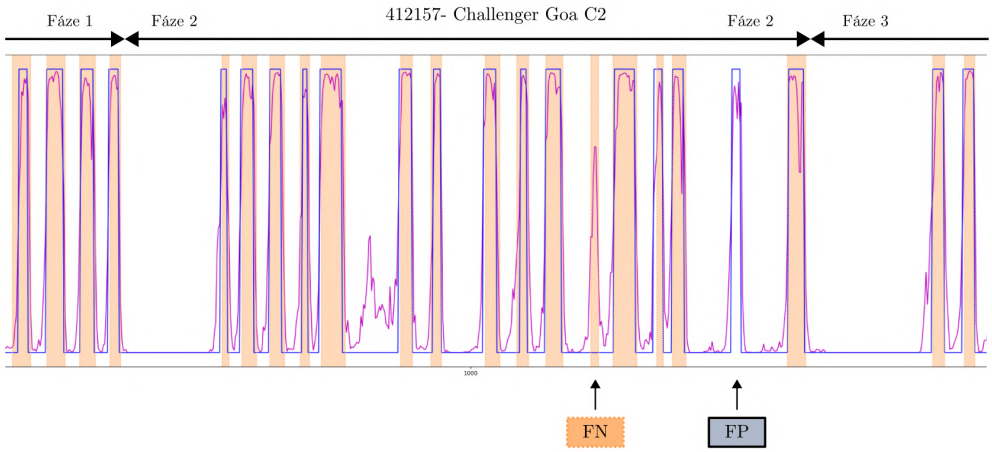
Model udržel kvalitu přesnosti klasifikace jednotlivých snímků nad 90 %. IoU skóre na datasetu z brazilského Recife dosáhlo pouhých 73,6 %, což je lehké zhoršení oproti výsledkům na druhém testovaném datasetu či validační/testovací množině původního datasetu. IoU skóre bylo sníženo kvůli větší nepřesnosti v detekci začátku a konce výměny. Průměrná absolutní chyba začátku výměny překonala hranici 2 sekund a chyba konce odpovídá 1,79 sekundě. Z pohledu úspěšnosti střihu algoritmus nastříhal zápasy z datasetu Doha 2024 v 73 % bez chyby. Toleranci 2 chyb algoritmus nepřekročil ani u jednoho zápasu, tudíž v toleranci nastříhal 100 % zápasů datasetu. Dataset Recife byl nastříhán v toleranci 2 chyb v 89 %. Pokud by se spojily výsledky střihu obou datasetů, tak model dokázal bezchybně nastříhat 68 % zápasů.

5.6 Kontrola výstupu časových značek

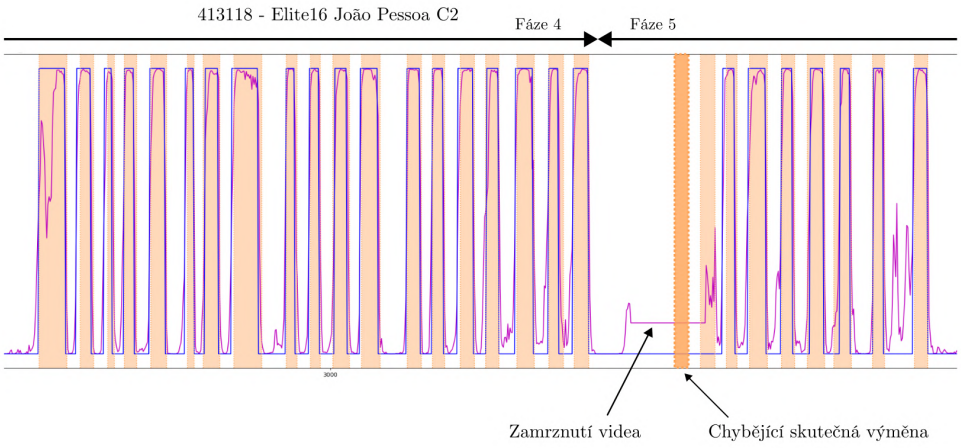
Algoritmus kontroly výstupů časových značek, jehož cílem je kontrola, zda predikovaný počet výměn odpovídá skutečnému počtu výměn, byl otestován na validační, testovací množině datasetu 1234. Testována byla úspěšnost správné kontroly. Kontrola je vyhodnocena jako správně provedená, pakliže počet FP a FN výměn získaných ze skutečných dat, je roven počtu přebývajících a počtu chybějících výměn získaný z kontrolního algoritmu. Kontrolní algoritmus využívá pouze informace o počtu výměn v každém setu, kdy nastaly oddechové časy a predikovaných časových značek, nikoliv skutečných časových značek výměn, které v praxi nebudou známy.

V testovací části datasetu chybí informace o počtu výměn a oddechových časech u jednoho zápasu – velikost testovací části pro kontrolu časových značek odpovídá 44 zápasům. Kontrola správně vyhodnotila 42 zápasů ze 44. Jedna chyba odpovídá situaci, kdy během jedné fáze byla vygenerována FP a FN výměna (Obrázek 5.9), druhá chyba je chybou vstupního videa a skutečných časových značek. Po technickém oddechovém čase zápasu (id: 413118) na centrálním kurtu turnaje Elite16 v João Pessoa, se zasekl obraz záznamu. Během zamrznutí obrazu lze podle píšťalky rozhodčího rozpoznat, že probíhá výměna, která není zachycena v časových značkách

skutečných dat, jelikož chybí obrazový podklad (Obrázek 5.10). Počet výměn dle konečného skóre se liší od počtu skutečných časových značek jednotlivých výměn. Pokud do datasetu není započítáno video s chybou skutečných dat, úspěšnost kontrolního algoritmu dosahuje 97,7 %.



Obrázek 5.9 Chyba kontrolního algoritmu na reálných datech – FP, FN.



Obrázek 5.10 Chyba kontrolního algoritmu na reálných datech – chybějící skutečná výměna.

Ve validační části datasetu chybí opět jeden zápas, tudíž bylo kontrolováno 56 z 57 zápasů datasetu. Kontrolní algoritmus správně vyhodnotil správnost či nesprávnost střihu (výskyt FP, FN výměn) u 54 zápasů. Chybná kontrola byla způsobena ve 2 případech, kdy byla vygenerovaná výměna navíc (FP), ale zároveň nebyla pokryta jedna skutečná výměna (FN) v rámci jedné fáze. Úspěšnost kontrolního algoritmu dosahuje 96,4% úspěšnosti.

Kapitola 6
Diskuze

Algoritmus automatického střihu beachvolejbalových videí využívající optical flow vykazuje nejhorší výsledky ze všech navržených algoritmů. Výsledky trpí na vysoký počet navíc vygenerovaných výměn (FP), zároveň ani nepokryje všechny skutečné výměny. Zisk a zakomponování časově závislé informace pomocí optical flow může vést k zpřesnění střihu, nikoliv však jako hlavní zdroj informace. Kombinací prostorové, obrazové a časové informace se zabývají 2 autorské kolektivy: Ng a kolektiv [17], K. Simonyan a A. Zisserman [16]. Algoritmy navržené v práci využívající model hlubokého učení je možné rozšířit o časovou informaci, nejen klasickými metodami, ale použitím dalšího modelu hlubokého učení – např. DeepFlow [32] či PWC-Net [33].

Algoritmy 2D klasifikace a 2.5D klasifikace pracují s videem jako se sekvencí jednotlivých snímků, které jsou uloženy ve formátu JPEG. Pro reálnou integraci automatického střihu do stávajícího softwaru je zapotřebí upravit načítání dat tak, aby vstupem nebyla sekvence vyextrahovaných snímků, nýbrž samotné video ve formátu MP4, a proces načítání paralelizovat. Export videa na sekvenci snímků je především paměťově náročný a navyšuje celkovou dobu zpracování jednoho videa o dobu extrakce snímků.

Výkonnost, úspěšnost modelu hlubokého učení je tak dobrá, jako je kvalita jeho trénovacích dat. Skutečná data o výměnách každého zápasu použitého v práci byla vytvořena různými skauty. Přestože jsou skauti školeni a znají definici, kdy začíná a končí beachvolejbalová výměna, reálná data vykazují rozptyl při zaznamenání začátku výměny. Začáteční časová značka výměny u některých zápasů označuje čas písknutí rozhodčího, u jiných dokonce až čas pár sekund před zahájením výměny podáním. Metrika průměrné absolutní chyby začátku výměny je kvůli tomuto trendu zatížena větší chybou. Konec výměny je ve videu jasněji a přesněji definován, jelikož prodleva mezi dopadnutím míče a ukončením výměny písknutím rozhodčího, je nepatrná. Experimenty napříč celou prací potvrzují trend větší chybovosti při začátku než při konci výměny.

Robustnost a vyšší úspěšnost střihu může být ovlivněna větším objemem dat či jemnější snímkovací frekvencí. Lokace a vzhled kurtů se během jedné sezóny mění, nicméně se často opakují lokace napříč jednotlivými sezónami. Obecně čím víc typů a druhů vzhledu kurtu model při tréninku uvidí, tím robustnější bude na nově příchozí data. Jen během experimentů proběhlo v aktuální beachvolejbalové sezóně 6 nových turnajů, které čítají více než 300 zápasů – více zápasů než velikost trénovací množiny. Zvýšení snímkovací frekvence může být provedeno globálně či na úrovni shluku. Globální zvýšení snímkovací frekvence zaručí modelu při tréninku, že uvidí stejnou sekundu výměny vícekrát. Zvýšením snímkovací frekvence na úrovni shluků předejde vícenásobné době tréninku, ale umožní zachytit jemnější časovou závislost. Počet shluků bude odpovídat počtu snímkům videa, nicméně shluk bude tvořen snímky s vyšší snímkovací frek-

vencí.

Algoritmus sice udržel kvalitu střihu videa při experimentu na datasetu dvou nových turnajů, nicméně může nastat situace, kdy další nový turnaj bude mít výrazně odlišný pohled z kamery a kvalitu záznamu než videa, na kterých byl model natrénován. V takovéto situaci bude mít algoritmus pravděpodobně problém udržet stejnou kvalitu střihu. Řešením by mohl být tzv. „finetuning“ na části odlišného datasetu, např. na záznamech z kvalifikační fáze turnaje, aby se zvýšil potenciál úspěšného střihu na záznamech z fáze hlavní soutěže turnaje.

Implementace 2D i 2.5D klasifikace byly postaveny na 2D deskriptoru architektury konvoluční sítě EfficientNet. Zvýšení přesnosti klasifikace a úspěšnosti střihu může zaručit změnu 2D architektury na 3D architekturu využívající 3D konvoluce, které jsou pro klasifikace videí využívány [18, 19]. Dále existuje potenciální možnost zlepšení kvality střihu postavením algoritmu na vision transformerech místo konvoluční neuronové sítě [46].

Závěr

Práce se soustředí na automatizovaný střih videa zápasů beachvolejbalu na jednotlivé výměny. Důležitou roli v úspěšném řešení problému sehrálo získání skutečných dat o časových značkách výměn. Časové značky jednotlivých výměn byly získány dvěma způsoby – mapováním již vystřižených výměn do původního videa a exportem časových značek z databáze. Obrazová data byla získána z FIVB databáze zápasů.

Byly navrženy a implementovány 3 algoritmy automatického střihu. První algoritmus využívá počítačového vidění bez procesu učení, konkrétně Farnerbackovy metody optical flow. Druhý a třetí algoritmus je postaven na principu hlubokého učení. Kostrou obou algoritmů je konvoluční neuronová síť EfficientNetV2-B3. Přístup 2D klasifikace zahazuje časovou souvislost mezi snímky videa a považuje vstupní video jako pole nezávislých snímků, čímž je problém klasifikace videa převeden na klasifikaci obrazu. 2.5D klasifikace – třetí navržený algoritmus – je rozšířením 2D klasifikace, kdy jsou konvoluční filtry roztažené do hloubky velikosti vstupních shluků a tím je do-klasifikace zanesena informace o časové závislosti.

Pro ohodnocení a porovnání algoritmů bylo prezentováno několik metrik soustředících se na úspěšnost klasifikace a úspěšnost z pohledu výměny a videa jako celku. Mezi metriky pro úspěšnost střihu patří IoU skóre, počet FP a FN výměn, chybovost časových značek v sekundách či precision a recall. Dále bylo hodnoceno, kolik videí jako celek algoritmus dokáže bez chyby nastříhat. Nejlépe hodnocený byl model 2.5D klasifikace složený ze dvou modelů, lišící se strukturou vstupního shluku.

Nejlépe hodnocený model byl podroben testem nového turnaje, jehož cílem bylo ověřit robustnost modelu na nových datech z praxe. Nad rámec zadání byl implementován kontrolní algoritmus, který vyhodnocuje výstupní časové značky výměn na základě informací o počtu výměn a oddechových časů.

Stanovené cíle byly splněny – byly vytvořeny datasety zápasů, které byly použity pro trénink a evaluaci navržených algoritmů, výsledky byly testovány, vizualizovány i diskutovány. Práce bude sloužit jako podklad pro integraci automatizace střihu videa do stávajícího produktu BeachData.

Příloha A

Rozdělení datasetů

Záznamy z beachvolejbalové sezóny 2023 turnajů Challenger a Elite16 byly rozděleny do 4 datasetů. Tabulky A.1, A.2 , A.3 , A.4 obsahují informaci o lokaci a typu turnaje, kurtu a zda byly záznamy použity pro trénink modelu. Celkový počet videí obsahuje součet trénovací, validační a testovací množiny datasetu. Pro každý dataset jsou zobrazeny pohledy kamery z každé položky tabulek – každé lokace, včetně kurtu (Obrázek A.1, A.2, A.3, A.4). Datasety pro experiment simulace reálného použití (Sekce 5.5) jsou definovány v tabulkách A.5, A.6) a pohledy jsou zobrazeny na obrázcích A.5, A.6.

Tabulka A.1 Záznamy patřící do skupiny č. 1.

Lokace	**Kurt**	**Použito při tréninku**
Ostrava, Elite16	C2	✓
Ostrava, Elite16	CC	✓
Gstaad, Elite16	CC	✓
Montreal, Elite16	C2	✓
Tepic, Elite16	CC	✓
Tepic, Elite16	C2	✓
Paříž, Elite16	C2	✓
Hamburg, Elite16	C2	✓
Hamburg, Elite16	CC	✓
Chiangmai, Challenger	CC	✓
Chiangmai, Challenger	C2	✓
Uberlandia, Elite16	C2	×
Uberlandia, Elite16	CC	×
João Pessoa, Elite16	CC	×
Celkový počet videí	$57 + 13 + 15 = 85$	


Ostrava, C2


Ostrava, CC


Gstaad, CC


Montreal, C2


Tepic, CC


Tepic, C2


Paříž, C2


Hamburg, C2


Hamburg, CC


Chiang Mai, CC


Chiang Mai, C2


Uberlandia, C2


Uberlandia, CC


João Pessoa, CC

Obrázek A.1 Pohledy z kamer – dataset č. 1.

Tabulka A.2 Záznamy patřící do skupiny č. 2.

Lokace	**Kurt**	**Použito při tréninku**
Doha, Finals	CC	✓
Doha, Finals	C2	✓
Edmont, Challenger	C2	✓
Edmont, Challenger	C3	✓
Nuvali, Challenger	CC	✓
Nuvali, Challenger	C3	✓
Haikou, Challenger	CC	✓
Jurmala, Challenger	CC	×
Goa, Challenger	CC	×
Goa, Challenger	C3	×
Goa, Challenger	C4	×
Celkový počet videí	$71 + 17 + 12 = 100$	

Doha Finals, CC | Doha Finals, C2 | Edmont, C2
Edmont, C3 | Nuvali, CC | Nuvali, C2
Haikou, CC | Jurmala, CC | Goa, CC
Goa, C3 | Goa, C4

Obrázek A.2 Pohledy z kamer – dataset č. 2.

Tabulka A.3 Záznamy patřící do skupiny č. 3.

Lokace	**Kurt**	**Použito při tréninku**
Montreal, Elite16	CC	✓
Paříž, Elite16	CC	✓
Nuvali, Challenger	C2	✓
Goa, Challenger	C2	x
Celkový počet videí	$34 + 9 + 4 = 47$	

Montreal, CC

Paříž, CC

Nuvali, C2

Goa, C2

Obrázek A.3 Pohledy z kamer – dataset č. 3.

Tabulka A.4 Záznamy patřící do skupiny č. 4.

Lokace	Kurt	Použito při tréninku
Gstaad, Elite16	C2	✓
João Pessoa, Elite16	C2	✓
Chiang Mai, Challenger	C3	✓
Espinho, Challenger	C2	✓
Espinho, Challenger	C3	✓
Espinho, Challenger	CC	×
Haikou, Challenger	C2	×
Haikou, Challenger	C3	×
Celkový počet videí	$59 + 18 + 15 = 92$	


Gstaad, C2


João Pessoa, C2


Chiang Mai, C3


Espinho, C2


Espinho, C3

Espinho, CC


Haikou, C2


Haikou, C3

Obrázek A.4 Pohledy z kamer – dataset č. 4.

Tabulka A.5 Záznamy patřící do datasetu Doha Elite16, 2024.

Lokace	**Kurt**	**Použito při tréninku**
Doha 2024, Elite16	CC	x
Doha 2024, Elite16	C2	x
Celkový počet videí	51	

Doha 2024, CC

Doha 2024, C2

Obrázek A.5 Pohledy z kamer – Doha Elite16, 2024.

Tabulka A.6 Záznamy patřící do datasetu Recife Challenger, 2024.

Lokace	**Kurt**	**Použito při tréninku**
Recife 2024, Challenger	CC	x
Recife 2024, Challenger	C2	x
Recife 2024, Challenger	C3	x
Recife 2024, Challenger	C4	x
Celkový počet videí	37	

Recife, CC

Recife, C2

Recife, C3

Recife, C4

Obrázek A.6 Pohledy z kamer – Recife Challenger, 2024.

Bibliografie

1. SRL, Genius Sports Italy. *Data Volley*. 2024. Dostupné také z: https://www.dataproject.com/Products/EN/en/Volleyball/DataVolley4.
2. CHENG, Xina; LIANG, Linzi; IKENAGA, Takeshi. Automatic data volley: game data acquisition with temporal-spatial filters. *Complex & Intelligent Systems*. 2022, roč. 8, č. 6, s. 4993–5010. ISSN 2198-6053. Dostupné z DOI: 10.1007/s40747-022-00752-3.
3. HARABAGIU, Neculai; PÂRVU, Carmen. The Statistical Analysis of the Game Actions of the Middle-Blocker Based on the Application of the "Data Volley" Software. *Revista Romaneasca pentru Educatie Multidimensionala*. 2022, roč. 14, č. 14, s. 101–110. Dostupné z DOI: 10.18662/rrem/14.1Sup1/539.
4. SOTIRIS DRIKOS Panagiotis Kountouris, Alexandros Laios; LAIOS, Yiannis. Correlates of Team Performance in Volleyball. *International Journal of Performance Analysis in Sport*. 2009, roč. 9, č. 2, s. 149–156. Dostupné z DOI: 10.1080/24748668.2009.11868472.
5. MIGUEL SILVA Tine Sattler, Daniel Lacerda; JOÃO, Paulo Vicente. Match analysis according to the performance of team rotations in Volleyball. *International Journal of Performance Analysis in Sport*. 2016, roč. 16, č. 3, s. 1076–1086. Dostupné z DOI: 10.1080/24748668.2016.11868949.
6. SRL, Genius Sports Italy. *Click&Scout*. 2024. Dostupné také z: https://www.dataproject.com/Products/US/en/Volleyball/ClickAndScout.
7. DATA, Beach. *Beach Data*. 2024. Dostupné také z: https://www.beach-data.com.
8. PERFORM, Stats. *Stats Perform*. 2024. Dostupné také z: https://www.statsperform.com.
9. ITAZURI, Takahiro; FUKUSATO, Tsukasa; YAMAGUCHI, Shugo; MORISHIMA, Shigeo. Court-Based Volleyball Video Summarization Focusing on Rally Scene. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, s. 179–186. Dostupné z DOI: 10.1109/CVPRW.2017.28.
10. KAWAMURA, Shunya; FUKUSATO, Tsukasa; HIRAI, Tatsunori; MORISHIMA, Shigeo. RSViewer: An Efficient Video Viewer for Racquet Sports Focusing on Rally Scenes. In: 2016, s. 247–254. Dostupné z DOI: 10.5220/0005670802470254.
11. ZHAO, Feng; DONG, Yuan; WEI, Zhe; WANG, Haila. Matching logos for slow motion replay detection in broadcast sports video. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012, s. 1409–1412. Dostupné z DOI: 10.1109/ICASSP.2012.6288154.

12. PEKER, Kadir A.; CABASSON, Romain; DIVAKARAN, Ajay. Rapid generation of sports video highlights using the MPEG-7 motion activity descriptor. In: YEUNG, Minerva M.; LI, Chung-Sheng; LIENHART, Rainer W. (ed.). *Storage and Retrieval for Media Databases 2002*. SPIE, International Society for Optics a Photonics, 2001, sv. 4676, s. 318–323. Dostupné z DOI: 10.1117/12.451102.

13. JEANNIN, S.; DIVAKARAN, A. MPEG-7 visual motion descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*. 2001, roč. 11, č. 6, s. 720–724. Dostupné z DOI: 10.1109/76.927428.

14. RAFIQ, Muhammad; RAFIQ, Ghazala; AGYEMAN, Rockson; JIN, Seong-Il; CHOI, Gyu Sang. Scene Classification for Sports Video Summarization Using Transfer Learning. *Sensors*. 2020, roč. 20, s. 1702. Dostupné z DOI: 10.3390/s20061702.

15. KARPATHY, Andrej; TODERICI, George; SHETTY, Sanketh; LEUNG, Thomas; SUKTHANKAR, Rahul; FEI-FEI, Li. Large-Scale Video Classification with Convolutional Neural Networks. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, s. 1725–1732. Dostupné z DOI: 10.1109/CVPR.2014.223.

16. SIMONYAN, Karen; ZISSERMAN, Andrew. *Two-Stream Convolutional Networks for Action Recognition in Videos*. 2014. Dostupné z arXiv: 1406.2199 [cs.CV].

17. NG, Joe Yue-Hei; HAUSKNECHT, Matthew; VIJAYANARASIMHAN, Sudheendra; VINYALS, Oriol; MONGA, Rajat; TODERICI, George. *Beyond Short Snippets: Deep Networks for Video Classification*. 2015. Dostupné z arXiv: 1503.08909 [cs.CV].

18. TRAN, Du; BOURDEV, Lubomir; FERGUS, Rob; TORRESANI, Lorenzo; PALURI, Manohar. *Learning Spatiotemporal Features with 3D Convolutional Networks*. 2015. Dostupné z arXiv: 1412.0767 [cs.CV].

19. HUANG, Yuzhong; BAI, Xue; WANG, Oliver; CABA, Fabian; AGARWALA, Aseem. Learning Where to Cut from Edited Videos. In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2021, s. 3208–3216. Dostupné z DOI: 10.1109/ICCVW54120.2021.00360.

20. FIVB. *Official Beach Volleyball Rules 2021-2024* [online]. FIVB, 2021. Dostupné také z: https://www.fivb.com/en/beachvolleyball/thegame_bvb_glossary/officialrulesofthegames.

21. AVP Beach Volleyball Stat Cheat Sheet. In: *AVP* [online]. © 2024 [cit. 2024-03-18]. Dostupné z: https://avp.com/news/avp-beach-volleyball-stat-cheat-sheet/.

22. PETER J. BROCKWELL, Richard A. Davis. Introduction. In: *Introduction to Time Series and Forecasting*. Springer Cham, 2016, s. 1–37. ISBN 978-3-319-29854-2. Dostupné z DOI: 10.1007/978-3-319-29854-2.

23. HYNDMAN, Rob J. Moving Averages. In: *International Encyclopedia of Statistical Science*. Ed. LOVRIC, Miodrag. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, s. 866–869. ISBN 978-3-642-04898-2. Dostupné z DOI: 10.1007/978-3-642-04898-2_380.

24. JUSTUSSON, BI. Median filtering: Statistical properties. *Two-dimensional digital signal prcessing II: transforms and median filters*. 2006, s. 161–196.

25. ROLAND FRIED, Jochen Einbeck; GATHER, Ursula. Weighted Repeated Median Smoothing and Filtering. *Journal of the American Statistical Association*. 2007, roč. 102, č. 480, s. 1300–1308. Dostupné z DOI: 10.1198/016214507000001166.

26. HYNDMAN, Robin John; ATHANASOPOULOS, George. *Forecasting: Principles and Practice*. 2nd. Australia: OTexts, 2018.

27. WANG, Zhou; BOVIK, A.C.; SHEIKH, H.R.; SIMONCELLI, E.P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*. 2004, roč. 13, č. 4, s. 600–612. Dostupné z DOI: 10.1109/TIP.2003.819861.

28. SONKA, Milan; HLAVAC, Vaclav; BOYLE, Roger. *Image processing, analysis, and machine vision*. 4. vyd. Cengage Learning, 2015. ISBN 1-133-59360-7.

29. HORN, Berthold K.P.; SCHUNCK, Brian G. Determining optical flow. *Artificial Intelligence*. 1981, roč. 17, č. 1, s. 185–203. ISSN 0004-3702. Dostupné z DOI: https://doi.org/10.1016/0004-3702(81)90024-2.

30. LUCAS, Bruce; KANADE, Takeo. An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI). In: 1981, sv. 81.

31. FARNEBÄCK, Gunnar. Two-Frame Motion Estimation Based on Polynomial Expansion. In: 2003, sv. 2749, s. 363–370. ISBN 978-3-540-40601-3. Dostupné z DOI: 10.1007/3-540-45103-X_50.

32. WEINZAEPFEL, Philippe; REVAUD, Jerome; HARCHAOUI, Zaid; SCHMID, Cordelia. DeepFlow: Large Displacement Optical Flow with Deep Matching. In: *2013 IEEE International Conference on Computer Vision*. 2013, s. 1385–1392. Dostupné z DOI: 10.1109/ICCV.2013.175.

33. SUN, Deqing; YANG, Xiaodong; LIU, Ming-Yu; KAUTZ, Jan. *PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume*. 2018. Dostupné z arXiv: 1709.02371 [cs.CV].

34. BISHOP, Christopher Michael; BISHOP, Hugh. *Deep Learning - Foundations and Concepts*. 1. vyd. Ed. CHAM, Springer. 2023. ISBN 978-3-031-45468-4. Dostupné z DOI: https://doi.org/10.1007/978-3-031-45468-4.

35. SANDLER, Mark; HOWARD, Andrew; ZHU, Menglong; ZHMOGINOV, Andrey; CHEN, Liang-Chieh. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2019. Dostupné z arXiv: 1801.04381 [cs.CV].

36. TAN, Mingxing; LE, Quoc V. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020. Dostupné z arXiv: 1905.11946 [cs.LG].

37. TAN, Mingxing; LE, Quoc V. *EfficientNetV2: Smaller Models and Faster Training*. 2021. Dostupné z arXiv: 2104.00298 [cs.CV].

38. SZEGEDY, Christian; VANHOUCKE, Vincent; IOFFE, Sergey; SHLENS, Jonathon; WOJNA, Zbigniew. *Rethinking the Inception Architecture for Computer Vision*. 2015. Dostupné z arXiv: 1512.00567 [cs.CV].

39. RADENOVIĆ, Filip; TOLIAS, Giorgos; CHUM, Ondřej. *Fine-tuning CNN Image Retrieval with No Human Annotation*. 2018. Dostupné z arXiv: 1711.02512 [cs.CV].

40. RUSSAKOVSKY, Olga; DENG, Jia; SU, Hao; KRAUSE, Jonathan; SATHEESH, Sanjeev; MA, Sean; HUANG, Zhiheng; KARPATHY, Andrej; KHOSLA, Aditya; BERNSTEIN, Michael; BERG, Alexander C.; FEI-FEI, Li. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*. 2015, roč. 115, č. 3, s. 211–252. Dostupné z DOI: 10.1007/s11263-015-0816-y.

41. RIDNIK, Tal; BEN-BARUCH, Emanuel; NOY, Asaf; ZELNIK-MANOR, Lihi. *ImageNet-21K Pretraining for the Masses*. 2021. Dostupné z arXiv: 2104.10972 [cs.CV].

42. LOSHCHILOV, Ilya; HUTTER, Frank. *Decoupled Weight Decay Regularization*. 2019. Dostupné z arXiv: 1711.05101 [cs.LG].

43. LOSHCHILOV, Ilya; HUTTER, Frank. *SGDR: Stochastic Gradient Descent with Warm Restarts*. 2017. Dostupné z arXiv: 1608.03983 [cs.LG].

44. BUSLAEV, Alexander; IGLOVIKOV, Vladimir I.; KHVEDCHENYA, Eugene; PARINOV, Alex; DRUZHININ, Mikhail; KALININ, Alexandr A. Albumentations: Fast and Flexible Image Augmentations. *Information*. 2020, roč. 11, č. 2. ISSN 2078-2489. Dostupné z DOI: 10.3390/info11020125.

45. BROWN, Gavin. Ensemble Learning. In: *Encyclopedia of Machine Learning*. Ed. SAMMUT, Claude; WEBB, Geoffrey I. Boston, MA: Springer US, 2010, s. 312–320. ISBN 978-0-387-30164-8. Dostupné z DOI: 10.1007/978-0-387-30164-8_252.

46. DOSOVITSKIY, Alexey; BEYER, Lucas; KOLESNIKOV, Alexander; WEISSENBORN, Dirk; ZHAI, Xiaohua; UNTERTHINER, Thomas; DEHGHANI, Mostafa; MINDERER, Matthias; HEIGOLD, Georg; GELLY, Sylvain; USZKOREIT, Jakob; HOULSBY, Neil. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. Dostupné z arXiv: 2010.11929 [cs.CV].

Obsah příloh

- `README.md` ..... stručný popis obsahu repozitáře
- `data` ..... adresář se skutečnými daty
- `demo` ..... adresář s konfigurací a daty pro demo
- `results` ..... adresář s výsledky, vizualizacemi
- `src`
  - `classification` ..... zdrojové kódy implementace – 2D, 2.5D klasifikace
  - `scripts` ..... zdrojové kódy implementace – ostatní
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `thesis.pdf` ..... text práce ve formátu PDF