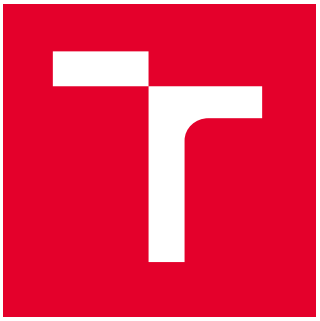


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

DETEKCE ANOMÁLIÍ VE VIDEOSEKVENCÍCH NA ZAŘÍZENÍCH S NÍZKÝM VÝPOČETNÍM VÝKONEM

ANOMALY DETECTION IN VIDEO SEQUENCES ON DEVICES WITH LOW COMPUTING POWER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — **František Bílek**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Pavel Sikora**

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

Student: František Bílek ***ID:*** 230534

Ročník: 3 ***Akademický rok:*** 2022/23

NÁZEV TÉMATU:

Detekce anomálií ve videosekvencích na zařízeních s nízkým výpočetním výkonem

POKYNY PRO VYPRACOVÁNÍ:

Student nastuduje a teoreticky popíše současné metody detekce anomálií (například odstranění/přidání/přemístění objektu) ve videosekvencích na zařízeních s nízkým výkonem, tedy na zařízeních disponujícími pouze procesorem s maximálně čtyřmi jádry. Vybrané metody implementuje ve vlastní samostatné aplikaci, která bude umožňovat načtení videa ke zpracování, detekci anomálií a vhodnou interpretaci výsledků.

DOPORUČENÁ LITERATURA:

[1] GONZALEZ, R. C.; WOODS R. E.: Digital Image Processing, Prentice Hall, New Jersey, 2002.

[2] OpenCV documentation. 2021 OpenCV, [cit. 2021-09-10]. Dostupné z: https://docs.opencv.org/4.5.3/

Termín zadání: 6.2.2023 ***Termín odevzdání:*** 26.5.2023

Vedoucí práce: Ing. Pavel Sikora

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

ABSTRAKT

Bakalářská práce se zaměřuje na problematiku detekce anomálií ve videosekvencích na zařízeních s nízkým výpočetním výkonem. Popsány jsou tradiční i současné přístupy detekce anomálií z hlediska strojového učení a neuronových sítí. Cílem práce je implementovat efektivní a spolehlivý algoritmus, který bude detekovat anomálie v reálném čase. Důraz je kladen na minimalizaci výpočetních nároků a optimalizaci paměťového využití, aby bylo dosaženo efektivity na zařízeních s omezenými výpočetními kapacitami.

KLÍČOVÁ SLOVA

Anomalib, detekce anomálií, FastFlow, neuronové sítě, PaDiM, strojové učení, videosekvence

ABSTRACT

The bachelor's thesis focuses on the problem of anomaly detection in video sequences on devices with low computational power. Traditional and current approaches to anomaly detection are described from the perspective of machine learning and neural networks. The goal of the thesis is to implement an efficient and reliable algorithm capable of detecting anomalies in real-time. Emphasis is placed on minimizing computational requirements and optimizing memory usage to achieve efficiency on devices with limited computational capacities.

KEYWORDS

Anomalib, anomaly detection, FastFlow, machine learning, neural networks, PaDiM, video sequence

BÍLEK, František. *Detekce anomálií ve videosekvencích na zařízeních s nízkým výpočetním výkonem*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023, 63 s. Bakalářská práce. Vedoucí práce: Ing. Pavel Sikora

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	František Bílek
VUT ID autora:	230534
Typ práce:	Bakalářská práce
Akademický rok:	2022/23
Téma závěrečné práce:	Detekce anomálií ve videosekvencích na zařízeních s nízkým výpočetním výkonem

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Pavlu Sikorovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Úvod

S poklesem finančních nákladů na veškerou elektroniku a rostoucí poptávkou po zabezpečení jsou do veřejných i soukromých prostor ve stále větší míře instalovány dohledové kamery. Jejich účelem je zachycení případných nekalých aktivit, zejména kriminální činnosti nebo vandalismu. Značný nárůst těchto sledovacích zařízení způsobuje, že množství zachyceného video obsahu předčuje lidské schopnosti jeho manuální analýzy. K anomálnímu jevu typicky dochází jen velmi vzácně v porovnání s délkou záznamu. Odhalování takových událostí člověkem je tedy značně neefektivní a často i nespolehlivá procedura. Z tohoto důvodu je v praxi maximální snaha o automatizaci tohoto procesu s využitím výpočetních technologií.

Klíčovou součástí problémů souvisejících s počítačovým viděním je extrakce atributů (feature extraction) ze vstupních dat. Jedná se o určité vlastnosti které mají výpovědní hodnotu pro řešený problém. Ve videosekvencích by tyto extrahované atributy měly být schopny zachytit rozdíl mezi normální a anomální událostí. Některé metody detekce anomálií využívají tzv. učení s učitelem, kdy učení probíhá na normálních i anomálních událostech, které jsou příslušně označeny. V oblasti detekce anomálií ve videosekvencích je ale více používané tzv. učení bez učitele. Model je v takovém případě trénován na datech, která obsahují pouze normální události. Jedním z hlavních důvodů je, že v praktických aplikacích běžně není k dispozici dostatek označených anomálních dat. Anomálii lze obecně definovat jako událost, která se jakýmkoliv způsobem odlišuje od normálního neboli predikovatelného chování určitého systému. V případě videosekvencí z dohledových kamer může anomálie typicky představovat poškozování objektu, výtržnictví či krádež.

Při automatickém zpracování obrazu a detekci anomálií je nezbytné se zabývat nároky na výpočetní výkon. Tato bakalářská práce se zaměřuje na zařízení s nízkým výpočetním výkonem, tedy zařízení, která nepoužívají hardwarovou akceleraci (např. GPU, Graphics Processing Unit) a disponují procesorem s maximálně čtyřmi jádry a malou fyzickou pamětí. S přihlédnutím k těmto požadavkům je tedy cílem implementovat efektivní a spolehlivý algoritmus, který bude schopen detekovat anomálie ve videosekvencích na zařízeních s nízkým výpočetním výkonem. Důraz je kladen na minimalizaci výpočetních nároků a optimalizaci paměťového využití, aby bylo možné provádět detekci v reálném čase s dostatečnou přesností.

1 Strojové učení

Detekce anomálií označuje proces identifikace anomálních složek ve vstupních datech. Jako hlavní nástroj pro detekci anomálií se v posledních letech stále více prosazuje strojové učení (Machine Learning, ML). Jedná se o vědní disciplínu, která se zabývá programováním počítačů za účelem jejich adaptace na lidské myšlení a uvažování. U detekce anomálií dochází k sestavení algoritmu, který získává znalosti s využitím trénovacích dat a je následně schopen rozlišovat mezi normálními a anomálními třídami [1]. Strojové učení pomáhá urychlit a usnadnit detekci anomálií a efektivně tak šetří lidské zdroje.

1.1 Typy strojového učení

Detekci anomálií lze z hlediska strojového učení rozdělit na 3 podkategorie podle funkce trénovacích dat - učení s učitelem, učení bez učitele a kombinace učení s učitelem a bez učitele.

1.1.1 Učení s učitelem

Učení s učitelem (supervised learning) je typ strojového učení, při kterém se k trénování modelu využívají data obsahující normální i anomální případy a každá instance z trénovací datové sady má své odpovídající označení (label). Cílem je vytvořit prediktivní modely pro anomálie a normální události, porovnat je mezi sebou a zvolit ten s nejvyšší přeností. Nevýhodou je, že četnost anomálií v trénovacích datech je velmi malá ve srovnání s běžnými událostmi. Vzhledem k časté neurčitosti a kontextuálnosti anomálií také vzniká problém při označování trénovacích dat [1]. Množství trénovacích dat nutné pro kvalitní natrénování modelu závisí na jejich kvalitě, náročnosti řešeného problému, či zvoleném algoritmu. [11]

Klasifikace

Klasifikační model strojového učení se používá k určení kategorie vstupních dat. Podle počtu výstupních kategorií se dělí na binární a vícetřídní klasifikátor. V problematice detekce anomálií ve videosekvencích se k predikci používá nejčastěji binární klasifikátor. Jednotlivé snímky nebo části videí mají přidružená označení podle normální a anomální činnosti. [1]

1.1.2 Učení bez učitele

K učení bez učitele (unsupervised learning) dochází, když jsou modelu poskytnuta pouze neoznačená vstupní data. Model v nich musí sám najít skryté struktury a vztahy a naučit se vzorce normálních aktivit. Na rozdíl od učení s učitelem zde není požadavek na úplnou znalost dat ze strany uživatele. [12]

Tato metoda předpokládá, že normální případy jsou v testovacích datech mnohem běžnější než anomálie. Pokud však tento předpoklad selže, může docházet k vysoké chybovosti. Mnoho technik kombinace učení s učitelem a bez učitele může být prováděno v režimu učení bez učitele, neoznačená vstupní data jsou tedy použita pro trénování. Taková adaptace předpokládá, že v testovacích datech se vyskytuje pouze malé množství anomálií, vůči kterým je model během trénování dostatečně robustní. [1]

1.1.3 Kombinace učení s učitelem a bez učitele

Metody detekce anomálií využívající kombinaci učení s učitelem a bez učitele (semi-supervised learning) detekují anomální aktivity trénováním modelu pomocí pouze slabě označených normálních datových instancí (weakly labeled data). Tato technika je pro účely detekce anomálií ve videosekvencích používaná velmi často, protože uplatňuje výhody jak učení s učitelem, tak i učení bez učitele. Kvůli dostupnosti pouze normálních trénovacích dat nebo videí bez anomálií je tato detekce poloautomatizovaná. Tyto detekční metody jsou nejčastěji modelovány pomocí hlubokých autoenkodérů a trénovány s dostatečným množstvím trénovacích dat obsahujících pouze běžné události, takže pro normální události vytvářejí minimální rekonkonstrukční chybu. V případě anomálních aktivit model vytváří naopak vysokou chybu rekonstrukce, čehož se využívá při detekci. [7]

1.2 Tradiční metody strojového učení

Tato část představuje tři základní metody, které jsou často používány v oblasti strojového učení a mají široké uplatnění při detekci anomálií v datech. Jedná se o metodu podpůrných vektorů, algoritmus k-nejbližších sousedů a rozhodovací stromy.

1.2.1 Metoda podpůrných vektorů

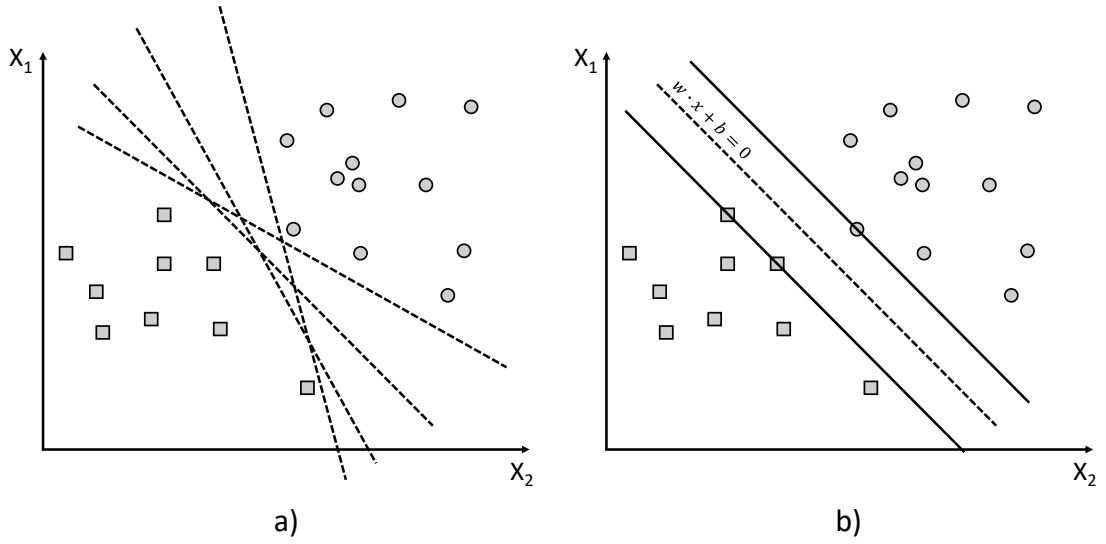
Trénovací metody nastavují parametry modelu takovým způsobem, aby byl každý vstupní vzorek správně klasifikován do své odpovídající třídy. Pokud je ale klasifikátor příliš přizpůsobený trénovacím datům, model si je začne pamatovat místo toho,

aby se naučil generalizovat a dochází tak k tzv. přeučení (overfitting). Metoda podpůrných vektorů (Support Vector Machine, SVM) umožňuje oddělit jednotlivé třídy trénovací množiny plochou, která maximalizuje okraj mezi nimi. Tím se zvyšuje schopnost generalizace modelu.

Pokud je uvažován nejjednodušší případ SVM, tedy lineárně separovatelná data v prostoru atributů, nadrovina (hyperplane), která data odděluje, je prostá přímka. Jak je naznačeno na obrázku 1.1, takových přímek může být nekonečně mnoho. Schopnost zobecnění závisí na lokalitě separační nadroviny a nadroviny s maximálním odstupem od nejbližších trénovacích vzorků. Tato nadrovina se nazývá nadrovina optimální separace a je určena rovnicí [2]:

$$\vec{w} \cdot \vec{x} + b = 0, \tag{1.1}$$

kde $\vec{w}$ je normála nadroviny, $\vec{x}$ je trénovací vzorek vyjádřený vektorem a b je posuv od počátku souřadnicového systému. Pro natrénování ideálního klasifikátoru je snaha o nalezení optimální nadroviny a dvou k ní paralelních nadrovin, jejichž vzdálenost je maximalizována. Vzniká tak pásmo na jehož okrajích se mohou vyskytovat datové body, které se nazývají podpůrné vektory (viz Obr. 1.1). [2]



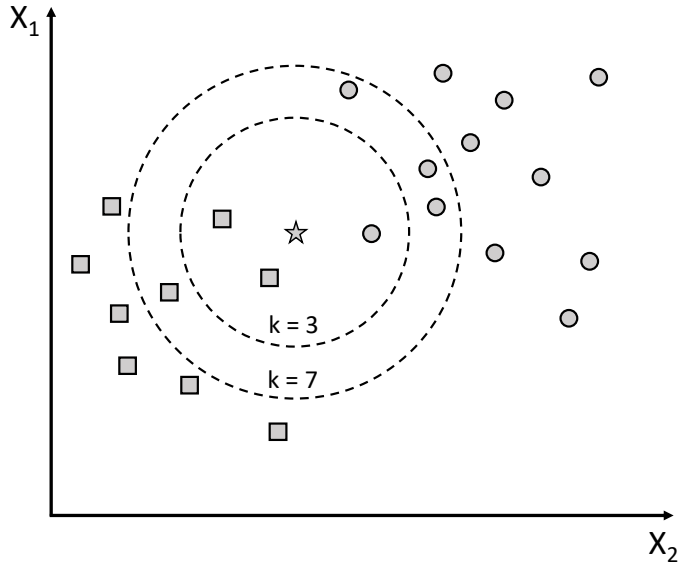
Obr. 1.1: Metoda podpůrných vektorů: a) různá umístění nadroviny, b) ideální klasifikátor

1.2.2 Algoritmus k-nejbližších sousedů

Algoritmus k-nejbližších sousedů (k-Nearest Neighbor, k-NN) je rychlá, efektivní a snadno implemetovatelná neparametrická technika klasifikace vzorků. Využívá databázi, ve které jsou data seskupena do několika tříd a algoritmus klasifikuje vstupní

vzorek, který je mu přidělen jako klasifikační problém [13]. Metody detekce anomálií využívající k-NN předpokládají, že normální datové instance se shlukují v hustých klastrech, zatímco anomálie se vyskytují daleko od svých k-nejbližších sousedů [17]. Míru anomálnosti lze vypočíst například pomocí Euklidovské, Manhattanské nebo Minkowskiho vzdálenosti. S rostoucí velikostí trénovací sady je algoritmus výpočetně náročný a je náchylný na případné nerelevantní vzorky ve vstupních datech, které mohou zapříčinit degradaci přesnosti [13].

Princip metody k-NN je naznačen na obrázku 1.2. Nachází se zde dvě třídy znázorněné čtverci a kolečky. Jedna z těchto tříd by mohla symbolizovat normální data a druhá anomálie. Předložený datový vzorek je znázorněn hvězdou. Pomocí jedné ze zmíněných vzdáleností je k němu nalezeno k nejbližších datových bodů (sousedů). Vzorek je následně klasifikován do třídy, které náleží maximum z těchto k sousedů.

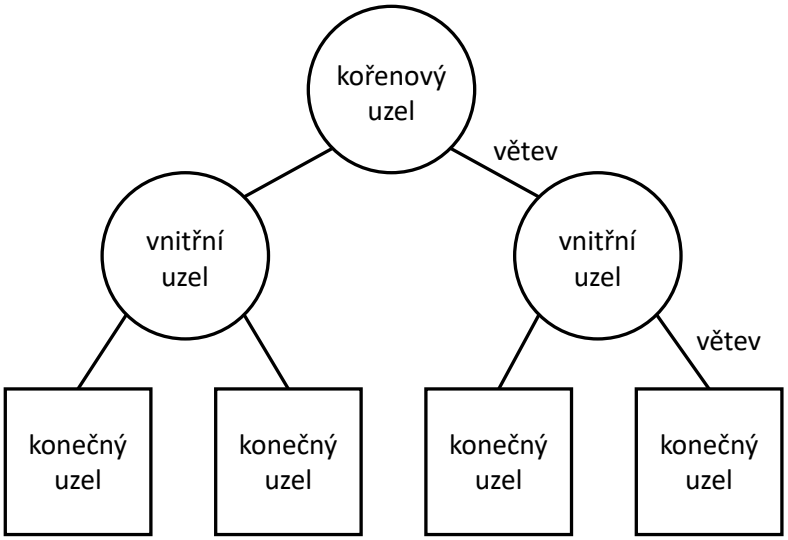


Obr. 1.2: Algoritmus k-nejbližších sousedů

1.2.3 Rozhodovací stromy

Tato metoda spočívá v konstrukci rozhodovacích stromů z trénovacích dat. Ve strojovém učení je rozhodovací strom (decision tree) také nazýván jako predikční model nebo klasifikační strom. Je to stromový graf, který je podobný struktuře vývojového diagramu (viz Obr. 1.3). Kořenový uzel (root node) reprezentuje celou množinu dat, vnitřní uzly (internal nodes) jsou testovací vlastností, každá větev (branch) představuje výsledek testu a konečné uzly neboli listy (leaf nodes) představují třídu, do které patří testovaný objekt. Rozhodovací stromy se používají k detekci anomálií,

protože jsou schopny identifikovat určité vzory, které vedou k anomáliím. Na základě rozhodovacího procesu stromu je možné odhalit anomálie ve vstupních datech na základě neobvyklých větví a rozhodovacích cest. [18]



Obr. 1.3: Rozhodovací strom

1.3 Zpětnovazební učení

Ve standardním modelu zpětnovazebního učení (reinforcement learning) se model (agent) rozhoduje podle aktuálního vstupu a prostředí, ve kterém se nachází. V každém kroku agent přijímá vstup s, tj. indikaci aktuálního stavu prostředí. Agent pak vybere akci a, která se vygeneruje jako výstup. Akce mění stav prostředí a hodnota tohoto přechodu stavu je sdělena agentovi prostřednictvím zpětnovazebního signálu r (odměna). Agent by měl volit takové akce, které dlouhodobě zvyšují jeho zisk z odměn. Lidská angažovanost je omezena pouze na změnu prostředí a vyladění systému odměn. Při snaze o maximalizaci odměny je agent náchylný hledat neočekávané způsoby, jak toho dosáhnout. Zapojení člověka spočívá v navádění stroje k provedení úkolu očekávaným způsobem. Zpětnovazební učení přináší v oblasti detekce anomálií výhody, ale zároveň vyžaduje dostupnost informací o anomálnosti, aby mohla být počítači poskytnuta kvalitní zpětná vazba. [12]

1.4 Reprezentační učení

Výkon modelů strojového učení je velmi závislý na reprezentaci dat (případně atributů), na které jsou aplikovány. Z toho důvodu jsou do návrhů ML algoritmů s vel-

kým úsilím zaváděny techniky předzpracování a transformace vstupních dat, jejichž výsledkem je jejich tzv. latentní reprezentace. Vymýšlením a vytvářením takových technik se zabývá oblast s názvem feature engineering. Za účelem rozšíření pole působnosti a usnadnění použitelnosti strojového učení je v současnosti však žádoucí, aby byly algoritmy strojového učení méně závislé právě na feature engineeringu. [15]

Myšlenka reprezentačního učení (representation learning) spočívá v automatizování procesu hledání vhodné reprezentace vstupních dat. V rámci získávání klíčových atributů ze vstupních dat model v určité míře přihlíží k dříve poskytnutým informacím o řešeném problému. Respektován je zde tzv. No Free Lunch Theorem, tedy, že nelze natrénovat univerzální model pro libovolný klasifikační úkol. Namísto toho je třeba vybírat a optimalizovat algoritmy podle konkrétního řešeného problému. [10]

U reprezentačního učení dochází ke snížení výpočetní náročnosti tím, že se rozměrná vizuální data redukují do nízkorozměrných vektorů. Metody reprezentace dat usnadňují získání užitečných informací při vytváření klasifikátorů nebo jiných prediktorů. Využití nachází kromě detekce anomálií i u dalších úkolů, jako je detekce a rozpoznávání objektů. Pro účely detekce anomálií ve videosekvencích byly vytvořeny metody založené na hledání globálních vzorů. Jsou jimi například model statického pozadí, optický tok nebo změna trajektorie (viz dále). [15]

1.5 Přenos učení

Tradiční pojetí strojového učení je charakteristické tím, že trénovací a testovací data mají stejný prostor atributů a stejné datové rozdělení. Pokud je mezi trénovacími a testovacími daty rozdíl v jejich distribuci, výsledky prediktivního modelu mohou degradovat. Zajištění dostatečného množství trénovacích dat, která dostatečně odpovídají prostoru atributů a předpokládaným charakteristikám distribuce testovacích dat může být obtížné nebo až nerealizovatelné. [14]

Přenos učení (Transfer Learning, TL) je podkategorie strojového učení při které se opětovně používají již existující modely v určité doméně k řešení nových problémů v související doméně. Tyto modely jsou již zcela nebo částečně natrénovány s využitím již existujících trénovacích dat, trénování vytvářeného prediktivního modelu tak nemusí probíhat zcela „od nuly“. Tento koncept se používá k účinnému zvýšení výkonu vytvářených prediktorů. Jak již bylo zmíněno, jedno z možných využití metody přenosu učení nastává tehdy, když není k dispozici dostatek označených trénovacích dat. To může být způsobeno tím, že data jsou vzácná nebo nedostupná či je jejich shromažďování a následné označování nákladné. S velkými datovými úložišti a již vytvořenými rozsáhlými datasety, které mají příbuznou (ale ne nutně totožnou) doménu s cílovými daty se stává z TL perspektivní přístup pro detekci anomálií. [14]

1.6 Few-Shot Learning

Strojové učení se zabývá konstruováním počítačových programů (modelů), které s přibývajícími zkušenostmi automaticky zlepšují svůj výkon. Typické aplikace strojového učení vyžadují pro učení s učitelem velké množšství označených trénovacích dat, což může v mnoha scénářích znamenat problém. Technika s názvem Few-Shot Learning (FSL) do jisté míry umožňuje trénování s využitím pouze omezeného množství trénovacích vzorků. Jedná se o speciální případ strojového učení, který se zaměřuje na získávání dobrého výkonu při učení z omezeného množšství označených dat poskytovaných v trénovací datové sadě. [16]

Kombinace učení s učitelem a bez učitele umožňuje pouze klasifikaci a regresi a používá kromě malého množšství označených dat hlavně neoznačená data. FSL využívá navíc zpětnovazebního učení a uvažuje různé druhy předchozích znalostí, jako jsou označená data z jiných domén nebo předem natrénované modely. Adaptace domény je druh přenosu učení, při kterém jsou zdrojová a cílová úloha stejné, ale domény, ve kterých jsou data získána, se liší. Metody přenosu učení jsou velmi často aplikovány ve FSL, kdy se předem natrénované modely adaptují na nový problém s omezeným množstvím dat. [16]

2 Umělé neuronové sítě

Umělé neuronové sítě (Artificial Neural Network, ANN) jsou rozsáhlým oborem v oblasti strojového učení a umělé inteligence. Tyto sítě jsou inspirovány biologickým nervovým systémem a slouží k modelování a simulaci chování lidského mozku. Základním stavením prvkem mozku je neuron nebo též nervová buňka. Synapse přenáší signály mezi neurony a umožňují tak jejich vzájemnou komunikaci. V umělých neuronových sítích je biologický neuron reprezentován umělým neuronem a synapse jsou zastoupeny váhami.

2.1 Dopředná neuronová síť

Jedná se o model umělé neuronové sítě s dopřednou topologií (Feed-forward Neural Network, FNN). Data zde proudí ze vstupu na výstup pouze jedním směrem bez zpětných smyček. Nejsou zde žádná omezení na počet vrstev, typ přenosové funkce nebo počet spojení mezi jednotlivými neurony. Nejjednodušší dopřednou neuronovou sítí je jednoduchý perceptron. [19]

Typická FNN je složena z neuronů, které jsou uspořádány do vrstev. První vrstva se nazývá vstupní, poslední vrstva se nazývá výstupní a vrstvy mezi jsou skryté. Mapovací funkce Γ_i přiřazuje neuronu i spojení ke všem neuronům přechozí vrstvy. Spojení mezi i-tým a j-tým neuronem je charakterizováno váhou w_{ij} a i-tý neuron má aktivační práh b_i. Aktivace i-tého neuronu je určena jako [20]:

$$x_i = f(p_i), \tag{2.1}$$

$$p_i = \sum_{j \in \Gamma_i} x_j w_{ij} + b_i, \tag{2.2}$$

kde p_i je potenciál i-tého neuronu a $f(p_i)$ je aktivační funkce. Koeficienty vah w_{ij} a prahů b_i jsou přizpůsobovány k nalezení minima součtu nejmenších čtverců rozdílů mezi vypočtenými a požadovanými výstupními hodnotami [20]:

$$E = \sum_o \tfrac{1}{2}(\mathbf{x}_o - \hat{\mathbf{x}}_o)^2, \tag{2.3}$$

kde $\mathbf{x}_o$ a $\hat{\mathbf{x}}_o$ jsou vektory vypočtených, respektive požadovaných aktivací neuronů výstupní vrstvy. Sčítá se přes všechny výstupní neurony o [20]. Funkce E se nazývá nákladová funkce (cost function).

2.1.1 Zpětná propagace

Algoritmus zpětné propagace je způsob učení s učitelem pro vícevrstvé dopředné neuronové sítě. V algoritmu zpětné propagace se používá iterační metoda nejstrmějšího klesání (steepest descent) pro hledání lokálního extrému nákladové funkce.

Hodnoty vah respektive prahů pro $(k+1)$-tou iteraci jsou [20]:

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} - \lambda \left(\frac{\partial E}{\partial w_{ij}} \right)^{(k)}, \tag{2.4}$$

$$b_{i}^{(k+1)} = b_{i}^{(k)} - \lambda \left(\frac{\partial E}{\partial b_{i}} \right)^{(k)}, \tag{2.5}$$

kde λ je míra učení (learning rate) [20]. Poloha následujícího bodu je určena derivací v aktuálním bodě, λ udává velikost kroku ve směru daném derivací. Získaná hodnota je následně odečtena od aktuální polohy. Tím dochází k minimalizaci funkce a tedy nalezení jejího lokálního minima.

2.1.2 Aktivační funkce

Neuron bez aktivační funkce provádí s využitím operací násobení a sčítání pouze lineární regresi vstupních dat. Aktivační funkce na neuron aplikuje nelineární transformaci, což dává neuronové síti schopnost řešit složitější problémy. Může se jednat o jakoukoliv matematickou funkci a její výběr se odvíjí od typu řešeného problému. V praxi jsou nejčastěji používané:

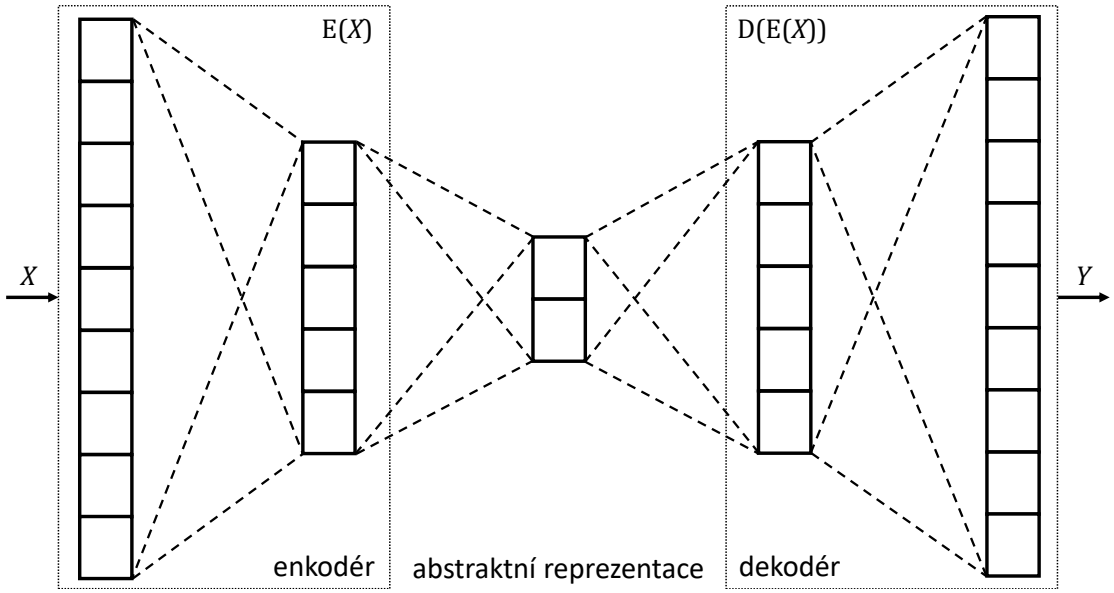
- Heavisideova funkce (jednotkový skok),
- usměrněná lineární funkce (Rectified Linear Unit, ReLU),
- hyperbolický tangens ($tanh(x)$),
- logistická funkce (sigmoida),
- normalizovaná exponenciální funkce (Softmax).

2.2 Autoenkodér

Autoenkodér je neuronová síť s dopřednou topologií, jejíž výstup vzniká kódováním a následným dekódováním vstupu a je trénovaná s využitím algoritmu zpětné propagace. Enkodér transformuje nezpracovaný vstup nebo jeho nízkoúrovňovou reprezentaci na vysokoúrovňovou abstraktní reprezentaci, ze které dekodér generuje rekonstruovanou verzi vstupu [6]. Cílem je minimalizovat chybu rekonstrukce. Mapovací funkce aplikuje mezi vstup a výstup určitou nelineární přenosovou funkci. Neuronová síť s více než jednou skrytou vrstvou se nazývá hluboký autoenkodér (Deep Auto Encoder, DeepAE). Autoenkodér lze formálně popsat jako [7]:

$$Y = D(E(X)), \tag{2.6}$$

kde X jsou data přivedená na vstup enkodéru, Y jsou rekonstruovaná vstupní data, E je funkce enkodéru, která kóduje vstupní data do skryté vrstvy a D je funkce dekodéru, která ze skryté vrstvy rekonstruuje data do výstupní vrstvy (Obr. 2.1).



Obr. 2.1: Základní struktura autoenkodéru

Cílem je natrénovat pár enkodér-dekodér tak, aby byla minimalizována chyba rekonstrukce $|X - Y|$ [7]:

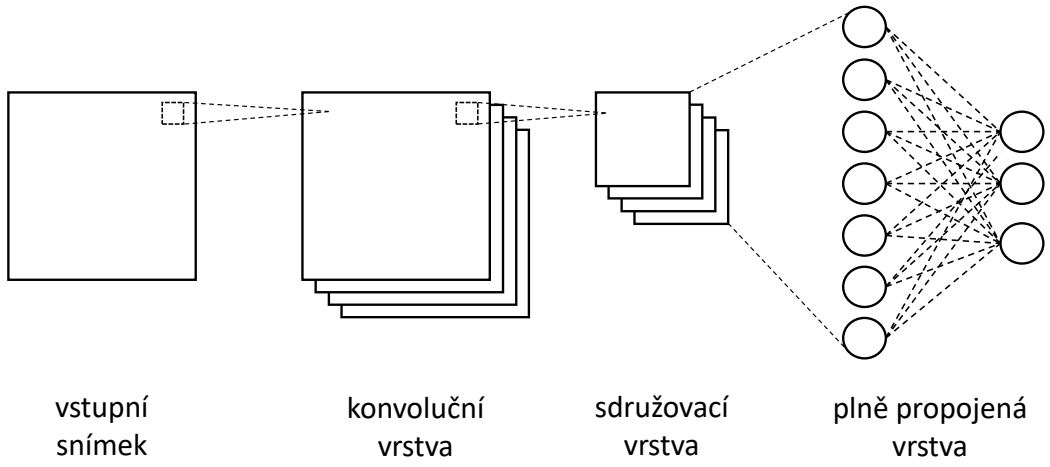
$$\min(|X - Y|) \tag{2.7}$$

2.3 Rekurentní neuronová síť

Rekurentní neuronová síť (Recurrent Neural Network, RNN) se nazývá rekurentní, protože provádí stejný úkol pro každý prvek datové sekvence, přičemž její výstup závisí na předchozích stavech. RNN může být zcela propojená nebo částečně propojená síť, včetně vícevrstvých dopředných sítí s odlišnými vstupními a výstupními vrstvami. Plně propojená síť nemá rozdělenou vstupní vrstvu a každý uzel (neuron) přijímá výstupy ze všech ostatních uzlů. Zpětná smyčka je možná pouze v rámci samotného neuronu [21]. RNN má tedy svým způsobem paměť, která zachycuje informace o dosavadních stavech [22]. Namísto tradičního algoritmu zpětné propagace (backpropagation) používají RNN pro výpočet gradientu algoritmus zpětné propagace v čase (Back Propagation Through Time, BPTT). Při zpětné propagaci model upravuje parametry tím, že vypočítá chyby od výstupní vrstvy k vstupní vrstvě. BPTT sčítá chybu v každém časovém kroku, protože RNN sdílí parametry napříč jednotlivými vrstvami [23]. Klíčovou vlastností rekurentních neuronových sítí je jejich schopnost uchovávat informaci pro pozdější použití v síti. Díky tomu jsou vhodné pro analýzu dat, která se mění v čase. Podkategorií rekurentních neuronových sítí jsou sítě Long Short Term Memory (LSTM). [22]

2.4 Konvoluční neuronová síť

Konvoluční neuronová síť (Convolutional Neural Network, CNN) je hojně využívaná architektura zejména v oblasti signálového a obrazového zpracování. Její název vyplývá z matematické lineární maticové operace zvané konvoluce. Nejpřínosnějším aspektem CNN je oproti obecné ANN princip sdílení vah, díky čemuž dochází ke značné redukci celkového počtu parametrů. S využitím CNN je možné vytvářet větší modely za účelem řešení složitějších úloh, což s klasickými ANN nebylo možné [24]. CNN je schopna efektivně získat klíčové vizuální atributy, které dobře reprezentují vstupní obraz. Rozpoznání vizuálních vzorů je možné dosáhnout i bez jakéhokoliv předzpracování vstupních dat. Základní struktura CNN je k vidění na obrázku 2.2. Tvořena je konvoluční (popsána v podsekci 2.4.1), sdružovací (podsekce 2.4.2) a plně propojenou vrstvou (podsekce 2.4.3) [25].



Obr. 2.2: Základní struktura CNN

V konvoluční neuronové síti již není každý neuron napojen na všechny neurony předchozí vrstvy, jako tomu bylo u FNN, ale pouze na malý počet neuronů. Váhy jsou obsaženy v malém konvolučním jádře (kernel), které je společné pro celou vstupní matici (snímek). Dochází tak k velmi účinému snižování počtu parametrů [26, 27]. Použití konvoluční neuronové sítě je však náročné na výpočetní výkon vzhledem k velkému počtu prováděných matematických operací [28].

2.4.1 Konvoluční vrstva

Účelem konvoluční vrstvy je získání atributů ze vstupních dat a uložit je do tzv. prostoru atributů (feature space). Tento proces se nazývá extrakce atributů (feature extraction). Atributy představují dílčí vlastnosti (rysy) daného obrázku, kterými

mohou být např. rovné čáry, rohy, okraje, ale i celé objekty. Prováděna je matematická operace konvoluce vyjádřena obecným vztahem [27]:

$$\mathbf{Y} = \mathbf{X} * \mathbf{W}, \tag{2.8}$$

kde $\mathbf{Y}$ je výstupní matice (prostor atributů), $\mathbf{X}$ je vstupní matice a $\mathbf{W}$ je konvoluční jádro. Vrstva používá k extrahování vlastností ze vstupu malý čtverec vstupních dat (patch), který je shodný s velikosti konvolučního jádra. Konvoluční jádro (kernel) je dvourozměrné pole, například o velikosti 3×3, obsahující váhy. Konvoluce vzniká postupným posouváním konvolučního jádra přes celou vstupní matici a výpočtem váženého průměru součinů hodnot v každém bodě [27]:

$$y[i,j] = (x * w)[i,j] = \sum_m \sum_n x[m,n] \cdot w[i-m, j-n], \tag{2.9}$$

kde i a j jsou souřadnice výstupního prostoru konvoluce, m a n představují relativní polohu vstupního prostoru konvoluce vzhledem k aktuální poloze na výstupu.

2.4.2 Sdružovací vrstva

Sdružovací (pooling) vrstva se používá k redukci a sjednocení prostoru atributů získaných konvoluční vrstvou. Na výstup konvoluční vrstvy se tak aplikuje další filtr. Parametr krok (stride) udává počet pixelů, o které se filtr po každém dílčím výpočtu posune. Výstupem je zmenšená matice, která ale zachovává klíčové vlastnosti ze vstupního snímku. Rozměr matice závisí na velikosti filtru, kroku a rozměrech vstupního snímku. Redukcí prostoru atributů sdružovací vrstvou se snižují výpočetní požadavky na samotnou neuronovou síť. V praxi se nejčastěji používají sdružení na základě [29]:

- Maximální hodnoty (max pooling),
- průměrné hodnoty (average pooling),
- součtu všech hodnot (sum pooling).

2.4.3 Plně propojená vrstva

Další používanou vrstvou v modelu CNN je plně propojená (fully-connected) vrstva. Může se jednat o jednu nebo více vrstev, které jsou tvořeny neurony. Všechny neurony v těchto vrstvách jsou vzájemně propojeny. Tyto vrstvy mají podobnou funkci jako skryté vrstvy v obecném modelu neuronové sítě. Účelem této vrstvy je z příznaků předchozích vrstev klasifikovat vstupní data na základě naučených charakteristik. [29]

2.5 Modulární neuronová síť

Současné metody považují jednotlivé snímky videa jako samostatné a nezávislé obrázky, což není vhodné pro detekci anomálií ve videosekvencích, kdy ve značné míře záleží na kontextu. Tyto metody jsou často kompletně naučeny na určitý typ anomálie. Ve velké míře využívají získané atributy ze vstupních snímků, které však nemusí vždy správně interpretovat danou událost. Je nereálné, aby tréninková data zcela pokryla všechny situace a způsoby chování, které mohou v rámci daného typu anomálie nastat. Proto by se neuronová síť měla být schopna dotrénovat na několika snímcích z nově poskytnutého videa (Few-Shot Learning), čímž proběhne adaptace na novou situaci a prostředí. V některých případech může být také požadováno, aby se neuronová síť byla schopna dynamicky přizpůsobit (nebo zobecnit) na zcela novou kategorii anomálií. [3]

3 Problematika detekce anomálií

Videosekvenci lze chápat jako komplexní systém, ve kterém dochází k mnoha různým transformacím (změna pozadí, osvětlení, hustoty davu apod.). Normální chování takového systému je pak popsáno statistickým modelem, který vychází z několika vybraných vzorků. V kontextu videosekvencí je takovým vzorkem například snímek nebo část videa. Tento statistický model umožňuje kvantifikovat normální třídu a odlišit ji od anomální třídy.

3.1 Anomálie ve videosekvencích

V současné době je běžné, že jsou veřejné i soukromé prostory monitorovány s využitím digitálních kamer. Tím vniká požadavek na jejich inteligentní dohled, čímž se zabývá oblast zvaná Analýza obsahu videa (Video Content Analysis, VCA). Rozpoznávání anomálních aktivit je jedním z užších zaměření, které hledá vzorce v chování a identifikuje události, které se liší od normálu. Rozpoznávání anomálních aktivit ve videích je také jedna ze zásadních problematik v oblasti počítačového vidění. Dohledové kamery mohou pokrývat prakticky neomeznou škálu prostředí (parkoviště, obchod, banka atd.), což představuje zásadní problém při rozpoznávání anomálních událostí. Vznikají totiž obrovské rozdíly uvnitř i napříč třídami, kdy třídy mají sice společné vlastnosti, ale jinak se zásadně liší. Dalším problémem je nedostatek označených anomálních dat, proměnlivé prostředí a nízké rozlišení videí. [4]

Anomálie obnáší celou škálu událostí od krádeže, dopravní nehody, vandalismu, přepadení apod. Tyto události se odehrávají na odlišných lokacích a během různé denní doby. Člověk rozpozná tyto anomálie na základě vlastních zkušeností a zdravého rozumu. Počítače identifikují takové případy na základě vizuálních atributů, které se naučily pomocí strojového učení. Algoritmy strojového učení fungují lépe, pokud jsou vstupní data uspořádaná a atributy jsou jednoznačně vázány na určitou třídu [4]. Důležité je také zmínit, že pojem anomálnost závisí na kontextu. Například auto jedoucí po silnici je obecně považováno za normální událost, ale v případě jízdy po chodníku se jedná o anomálii. Obecně lze anomálii definovat jako neočekávanou událost vyskytující se ve značně menší míře než událost normální [5].

3.2 Zařízení s nízkým výpočetním výkonem

Detekce anomálií na zařízeních s nízkým výpočetním výkonem je náročný problém, jelikož omezený výpočetní výkon a malá velikost paměti do značné míry omezují implemetaci komplexních algoritmů strojového učení pro detekci anomálií. Použití algortimů hlubokého učení (Deep Learning, DL) na málo výkonných zařízeních může

tedy být značně neefektivní, kvůli jejich omezeným výpočetním zdrojům. Například hluboké neuronové sítě (Deep Neural Network, DNN) a konvoluční neuronové sítě běžně používají sítě obsahující tisíce vzájemně propojených neuronů a až miliony parametrů [32]. Takové sítě jsou implementovány s využitím frameworků, které v základu využívají optimalizační knihovny pro zvýšení výpočetního výkonu procesorů a grafických karet (např. PyTorch CUDA) [30].

Jedním z přístupů pro detekci anomálií je použití cloudových služeb, které umožňují přenesení vysoké výpočetní zátěže na vzdálený server a zařízení s nízkým výpočetním výkonem je zodpovědné pouze za odesílání dat a přijímání výsledků. Tato metoda umožňuje použití komplexnějších algoritmů pro detekci anomálií [31]. Přenesení provádění inference do cloudu je sice přirozené řešení, ale nemusí být praktické vzhledem k případným výpadkům v síťové komunikaci. Není zaručeno, že cloudové přenosy a tedy aplikace samotná budou vždy dostupné. To může být kritické zejména v real-time scénářích [32].

Provedení klasifikace na CPU (Central Processing Unit) lokálního zařízení je pro velkou část existujících DL modelů nerealizovatelné. V některých scénářích je klasifikace možná prostřednictvím ručně vytvořeného DNN modelu s nízkými nároky na výpočetní výkon a paměťový prostor. Takový přístup ale vyžaduje vysokou míru úsilí a dovednosti při vytváření nových algoritmů [32]. Určitým kompromisem je použití výpočetně efektivních algoritmů strojového učení, které jsou speciálně navrženy pro implementaci na zařízeních s nízkým výpočetním výkonem. Tyto algoritmy jsou méně přesné, ale mohou být nasazeny na méně výkonných zařízeních [31].

3.3 Metody detekce anomálií

Detekce anomálií je v současných přístupech nejčastěji docíleno s využitím metod založených na DL. Nejdůležitějším aspektem detekce anomálií je reprezentace atributů. V DL metodách není získávání klíčových atributů záležitostí ruční práce člověka, ale automatického učení modelu na základě velkého množství dat. Pro tento úkol není vhodné model strojového učení trénovat metodou učení s učitelem kvůli značné disproporci mezi počtem normálních a anomálních datových instancí. Z tohoto důvodu se modely detekce anomálií často trénují s využitím pouze jedné třídy obsahující pouze normální případy a anomálie se během trénování modelu neuvažují. Při testování a inferenci natrénovaného modelu jsou data, která se liší od normálního trénovacího datasetu, klasifikována jako anomální [38]. Obecně se k detekci anomálií ve videosekvencích přistupuje velmi obdobně napříč různými metodami. Liší se často pouze v dílčích krocích a v jejich pořadí.

3.3.1 Analýza trajektorie objektu

V případě metod založených na trajektorii je objekt na videu detekován, sledován napříč snímky a generována je jeho trajektorie. Případná anomální činnost je pak odvozena z analýzy trajektorie [7]. Metody automatické detekce anomálií na základě trajektorií objektů přitahují v poslední době velkou pozornost, především díky rostoucímu množství dostupných historických dat. Navíc současný trend vývoje vede ke stále pokročilejším senzorovým systémům produkujícím obrovské množství dat o trajektoriích pohybujících se objektů, jako jsou lidé, vozidla nebo zvířata. Kvůli již zmíněnému dramatickému zvýšení množšství dohledových videokamer bylo vynaloženo značné úsilí směrem k výzkumu a vývoji algoritmů pro detekci a sledování objektů [8]. Účinnost těchto metod přímo závisí na úspěšnosti detekce objektu. Ta je ovlivněna především mírou zahuštění ostatními objekty, ale i dalšími parametry, jako rozlišení videa, změna počasí nebo změna světelných podmínek. Použití je tedy omezeno na málo proměnná prostředí s nízkou mírou zahuštění ostatními objekty. Navíc zde mohou chybět podstatné informace o kontextu videa [7].

3.3.2 Hledání globálních vzorů

Metody založené na hledání globálních vzorů získávají důležité atributy z videosekvence na různých sémantických úrovních pomocí výpočetně efektivních algoritmů. Nejčastěji se jedná o prostorové časové gradienty (spatial-temporal gradients) nebo optický tok (optical flow). Videosekvence je analyzována jako celek, není zde již tedy snaha o detekci a sledování samostatných objektů, jako je tomu u metod založených na trajektorii. Metody založené na globálních vzorech se ukázaly jako účinné pro středně i hustě zaplněná prostředí kvůli absenci detekce a sledování jednotlivých objektů. Mohou být velice účinné pro detekci anomálií v rámci videa, ale příliš se nehodí pro jejich lokalizaci. Metody, které implementují hledání globálních vzorů, jsou např. PCA (Principal Component Analysis) nebo model Gaussovy regrese. [9]

3.3.3 Řídké kódování

Řídké kódování (sparse coding) je často používanou technikou pro detekci anomálií ve videosekvencích. Využívá předpokladu, že lineární kombinace vzorů normálního chování dobře reprezentují normální aktivity s minimální chybou rekonstrukce. Model se učí na trénovacích videích, která obsahují pouze normální události. Anomální případy aplikované při inferenci tedy nebudou odpovídat natrénovanému modelu, což se projeví velkou chybou rekonstrukce. Problém při implementaci této techniky může spočívat v detekci ve videu obsahujícím více objektů a ve schopnosti lineárního

modelu, který je založený na řídkosti, správně oddělit třídy a účinně tak reprezentovat anomálie. Tyto problémy řeší adaptivní řídké kódování respektive nelineární modely. [7]

3.3.4 Prediktivní modely

Video lze chápat jako časoprostorový signál, kde konkrétní uspořádání snímků vytváří určitý vzor. V případě prediktivního modelování je cílem modelovat podmíněné rozdělení, což umožňuje předpovědět aktuální snímek na základě předchozích snímků. Prediktivní (nebo časoprostorové) modely jsou široce využívány pro detekci anomálií ve videosekvencích, protože kombinují jak prostorové (vizuální) tak časové (pohybové) vlastnosti. [7]

3.3.5 Generativní modely

V generativním modelování je snaha o naučení se pravděpodobnosti různých kombinací vstupních dat (X) a jejich příslušných výstupů (Y). Hlavním cílem je zjistit, jaká je pravděpodobnost, že dané vstupy patří k určitému výstupu. Hluboké generativní modely umožňují natrénovat pravděpodobnostní model na základě principu maximální věrohodnosti. To znamená, že je hledán model, který nejlépe zobecňuje pozorovaná data. Hluboké generativní modely se často používají v oblasti detekce anomálií ve videosekvencích kvůli jejich schopnosti řešit problémy s nedostatkem a nerovnováhou dat. [7]

3.3.6 Hluboké klasifikační modely

Vývoj pokročilých DL modelů pro detekci anomálií ve videu je náročný vzhledem k nedostatku přesně anotovaných anomálních dat a nejasné povaze samotných anomálií. Problém detekce anomálií bez přítomnosti označených trénovacích anomálních instancí se nazývá jednoduchá klasifikace (One Class Classification, OCC). Zabývá se již popsanou problematikou hledání hyperplochy nebo hypersféry, která maximalizuje vzdálenost od nejbližších normálních dat a minimalizuje překryv s anomáliemi. Při kombinaci s DNN, které mají schopnost hierarchického zpracování rysů vznikají tzv. Deep OCC modely. Tyto modely se na základě dosavadních studií ukázaly jako účinné pro detekci anomálií ve videosekvencích, nicméně vyžadují vyšší výpočetní výkon a delší dobu trénování než předchozí metody. [7]

3.4 Vyhodnocovací metriky

Detekce anomálií ve videosekvencích je záležitost binární klasifikace. Pokud jsou anomálie detekovány na úrovni snímků, je každé rozhodnutí klasifikátoru jednou z následujících čtyřech možností:

- Skutečně pozitivní (True Positive, TP) - anomální snímek vyhodnocen jako anomální,
- skutečně negativní (True Negative, TN) - normální snímek vyhodnocen jako normální,
- falešně pozitivní (False Positive, FP) - normální snímek vyhodnocen jako anomální,
- falešně negativní (False Negative, FN) - anomální snímek vyhodnocen jako normální.

Pro úspěšnou detekci anomálií je důležité minimalizovat především počet FN.

3.4.1 Specificita, senzitivita, skóre F1

Specificita (precision) je podíl správně vyhodnocených pozitivních případů ku celkovému počtu pozitivně vyhodnocených případů (viz rovnice 3.1). Senzitivita (recall) je podíl správně vyhodnocených pozitivních případů ku celkovému počtu pozitivních případů (viz rovnice 3.2). Skóre F1 (F1 Score) je harmonický průměr senzitivity a specificity a je dáno rovnicí 3.3. [29]

$$\text{Specificita} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{3.1}$$

$$\text{Senzitivita} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3.2}$$

$$\text{Skóre F1} = 2 \cdot \frac{\text{Specificita} \cdot \text{Senzitivita}}{\text{Specificita} + \text{Senzitivita}}. \tag{3.3}$$

3.4.2 ROC, AUROC

Křivka ROC (Receiver Operating Characteristics) je grafický nástroj používaný pro vyhodnocení výkonnosti klasifikačních modelů. Křivka je tvořena postupnou změnou prahové hodnoty pro rozhodnutí o klasifikaci a pro každou prahovou hodnotu se vypočítá úspěšnost klasifikace vzhledem k míře pravdivé pozitivity (True Positive Rate, TPR) a míře falešné pozitivity (False Positive Rate, FPR). ROC poskytuje přehled o výkonu klasifikačního modelu při různých prahových hodnotách a umožňuje porovnávat různé modely nebo optimalizovat nastavení modelu pro dosažení požadovaných výsledků. Plocha pod křivkou ROC (Area Under ROC, AUROC) je

často používanou metrikou pro vyhodnocení výkonnosti modelu, kde vyšší hodnota AUROC indikuje jeho lepší klasifikační schopnosti. [34]

4 Návrh aplikace

Cílem této práce je vytvoření aplikace pro detekci anomálií ve videosekvencích na zařízeních s nízkým výpočetním výkonem. Algoritmus bude zvolen tak, aby aplikace byla schopna provést celý proces na zařízeních s maximálně čtyřmi fyzickými jádry, bez technologie současného vícevláknového zpracování (Simultaneous Multi-Threading, SMT). Uvažován nebude ani případný hardwarový akcelerátor, jako GPU, TPU (Tensor Processing Unit) a podobné.

Na základě výše popsaného postupu bude vytvořen algoritmus, který by měl být schopen načíst video, rozdělit jej na snímky, provést dotrénování předtrénovaného modelu na několika snímcích z nového videa a následně otestovat zbylé snímky na výskyt anomálie.

4.1 Výběr metody detekce anomálií

V článku [35] jsou porovnány tradiční metody ML (jako zástupce je zvolena metoda SVM) s DL metodami. Hlavní přednost metod DL je, že jsou schopny automaticky zpracovávat dvourozměrná obrazová data a extrahovat z nich klíčové atributy. Tradiční ML metody vyžadují převod z dvourozměrných vektorů na jednorozměrné vektory. U DL modelů nedochází k chybné klasifikaci snímků kvůli případné podobnosti rysů napříč třídami. To naznačuje, že pro práci s vizuálními daty je DL vhodnější než tradiční metody ML.

Podle [36] přináší použití DL pro detekci anomálií několik výhod. Tyto přístupy jsou navrženy tak, aby efektivně pracovaly s proměnnými a vysoce rozměrnými daty, jako jsou videosekvence. DL umožňuje efektivně modelovat složité nelineární vztahy v datech a využít je pro úlohu detekce anomálií. Díky tomu je možné shromažďovat data z různých nezávislých zdrojů a není nutné se zabývat navrhováním složitých algoritmů pro jednotlivé typy anomálií. Výkon DL modelů se může potenciálně škálovat s dostupností trénovacích dat, což u tradičních metod strojového učení obecně neplatí. DL modely vyžadují pouze minimální doladění hyperparametrů za současného dosažení dobrých výsledků klasifikace.

Trénování složitých modelů neuronových sítí na velkých datech, jako jsou obrázky nebo videa, vyžaduje značný výpočetní výkon. Na zařízeních s omezenými prostředky, zejména s nízkou výpočetní kapacitou procesoru a malou pamětí RAM (Random Access Memory), je toto trénování časově náročné a v některých případech i nemožné. Proto je vhodné model natrénovat na výkonnějším počítači a následně provést dotrénování modelu na zařízení s nízkým výpočetním výkonem. Toto dotrénování se provádí pouze na malém množství snímků, aby se minimalizovala výpočetní náročnost. Tímto přístupem je možné využít přednosti předtrénovaného

modelu a současně snížit nároky na výpočetní prostředky zařízení s omezeným výkonem.

4.2 Knihovna Anomalib

Anomalib je knihovna určená pro práci s DL algoritmy, vytvořená vývojáři společnosti Intel. Jejím účelem je shromáždění nejmodernějších modelů pro detekci a lokalizaci anomálií a jejich srovnání na veřejných i privátních datasetech. Nabízí snadnou implementaci vybraných modelů a poskytuje sadu nástrojů pro vývoj vlastních modelů. Knihovna je zaměřena na práci s vizuálními daty. Podle dokumentace jsou jako vstupní data kromě obrázků podporována i videa, v současnosti ale pouze na úrovni jednotlivých snímků. Anomalib obsahuje algoritmy pro detekci anomálií, které dosahují dobrých výsledků na různých testovacích datasetech a lze je použít bez dalších úprav. K dispozici jsou nástroje jako logování experimentů, vizualizéry a optimalizátory hyperparametrů, které usnadňují implementaci modelů pro detekci anomálií. [37]

Dostupné modely ve verzi knihovny 0.3.7 jsou CFlow, DFKD, DFM, FastFlow, GANomaly, PaDiM, PatchCore a STFPM. Zmíněné modely jsou implementovány knihovnou PyTorch, která podporuje provádění všech operací na GPU. Anomalib umožňuje modulárně vytvářet vlastní algoritmy s využitím již hotových komponentů. Knihovna také poskytuje rozhraní pro nasazení modelů v reálném čase na GPU s využitím PyTorch nebo CPU pomocí OpenVINO. [37]

Číselný rozsah hodnoty anomaly score na úrovni snímků se může během inference lišit v závislosti na daném modelu a datové sadě. Za účelem převodu nezpracovaných anomaly score do standardizovaného formátu Anomalib normalizuje tyto hodnoty na rozsah od 0 do 1. Standardně Anomalib používá tzv. min-max normalizaci s ohledem na hodnoty pozorované během validace, ale metodu normalizace lze změnit (nebo zcela vyřadit) v konfiguračním souboru. [37]

4.3 Výběr modelů

Modely v Anomalib používají pro extrahování atributů architektury pátěřních hlubokých CNN ResNet18 nebo Wide ResNet-50-2. Jedná se o tzv. reziduální sítě, které se vyznačují inovativním přístupem pomocí tzv. reziduálního spojení. Podle typu použité páteřní sítě se odvíjí i celkový počet parametrů modelu. Páteřní sítě pocházejí z kolekce PyTorch Image Models (timm). [37]

V tabulce 4.1 jsou uvedeny počty parametrů pro jednotlivé modely včetně jejich celkové velikosti. Zařízení s nízkým výpočetním výkonem obvykle nedokážou efek-

tivně pracovat s modely s velkým počtem parametrů (20 M a více). Pokud je model příliš velký, může dojít k výraznému zpomalení procesu detekce anomálií nebo k přetížení zařízení. Proto byly pro prvotní testování vybrány modely s malým počtem parametrů. Tím lze zajistit výkonnost i na takových zařízeních, kde není k dispozici dostatečný výpočetní výkon.

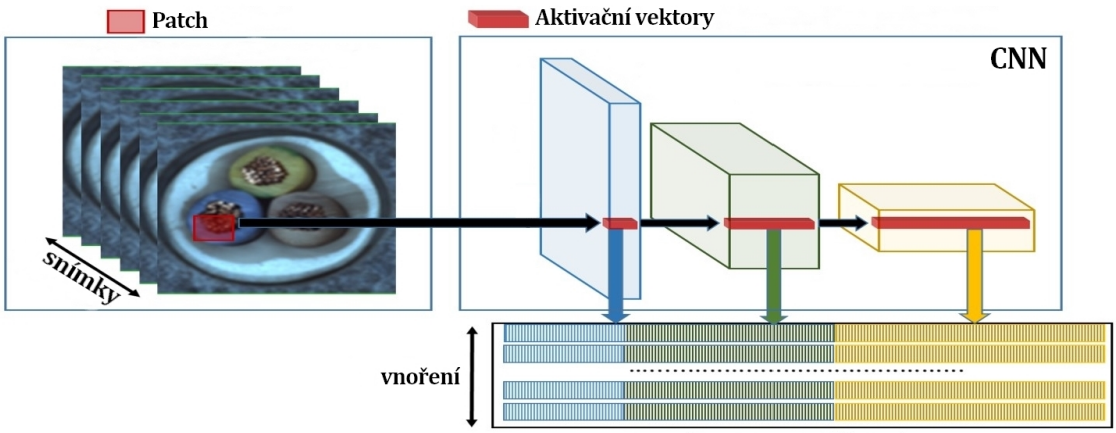
Tab. 4.1: Počet trénovatelných parametrů a jejich velikost

Model	**Počet parametrů**	**Velikost [MB]**
CFlow	236 M	946,249
GANomally	49,5 M	198,095
PatchCore	24,9 M	99,45
DFKD	11,2 M	44,706
FastFlow	5,7 M	39,567
STFPM	5,6 M	22,262
DFM	2,8 M	11,131
PaDiM	2,8 M	11,131

Modely byly vybírány podle jejich výpočetní náročnosti a paměťových nároků. Dále byla posouzena jejich vhodnost pro detekci anomálií ve videosekvencích podle dostupné dokumentace. Na základě těchto kritérií se ukázaly jako vhodné modely PaDiM (Patch Distribution Modeling Framework) a FastFlow. Jejich základní princip fungování je popsán v podsekcích 4.3.1, respektive 4.3.2. Předběžnými experimenty bylo zjištěno, že zbylé nízkoparametrické modely (DFKD, STFPM, DFM) mají po natrénování na vlastním datasetu (popsaném v podsekci 4.4) velmi malou přesnost. Naopak vybrané modely jsou po natrénování schopny požadované anomálie detekovat a byly tedy implementovány ve výsledné aplikaci.

4.3.1 Model PaDiM

Základní princip modelu PaDiM je na obrázku 4.1. Vstupní snímek je rozdělen na malé skupiny pixelů (patch). Model využívá natrénovanou vícevrstvou CNN pro získání atributů ze vstupního obrázku a jejich vnoření (embedding) do nízkorozměrných vektorů. Aktivační vektory z různých vrstev jsou zřetězeny tak, aby výsledné vektory nesly informace z různých sémantických úrovní. Tímto způsobem se zakódují jemné detaily i globální kontexty daného snímku. Hodnota anomaly score je vzdáleností mezi vektory vnoření testovaného snímku a referenčními vektory, které reprezentují normální stav z trénovacího datasetu. [38]



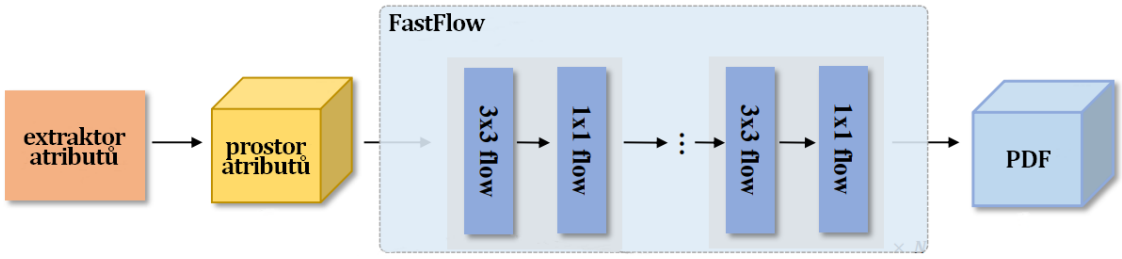
Obr. 4.1: Model PaDiM (převzato z [38] a upraveno)

Přesnost modelu PaDiM byla při vývoji vyhodnocena kromě standardního referenčního datasetu MVTec (s přesností 97,9 %) i na neuspořádaných datasetech jako ShanghaiTech Campus (STC). Výsledky naznačují, že model je aplikovatelný i na takto realistická data, jako jsou videosekvence z dohledových kamer. PaDiM je také vhodný z hlediska časové náročnosti a nároky na paměť. [38]

4.3.2 Model FastFlow

Model FastFlow využívá 2D normalizující tok pomocí páteřních CNN pro detekci anomálií na různých sémantických úrovních. Všechny tyto páteřní sítě jsou inicializovány s předem trénovanými váhami na databázi ImageNet a jejich parametry jsou pro následující trénovací proces zmrazeny. Trénování modelu probíhá pomocí optimalizátoru Adam. Nejprve jsou získány vizuální vlastnosti pomocí extraktoru příznaků, které následně vstupují do modulu FastFlow, kde dochází k odhadu hustoty rozdělení pravděpodobnosti (Probability Density Function, PDF). Ve fázi trénování je model trénován na normálních snímcích, aby vytvořil standardní normální 2D distribuci. Při inferenci jsou pro výpočet hodnoty anomaly score použity pravděpodobnosti každého umístění ve 2D rovině. [39]

Model FastFlow je podle [39] ve srovnání s CFlow, který používá podobnou techniku detekce, rychlejší a nabízí značné snížení parametrů. FastFlow využívá end-to-end inferenční fázi která má vysokou účinnost z perspektivy dodatečného času pro inferenci. Tato metoda dosahuje až 4násobného snížení inferenčního času oproti modelu CFlow a 10násobného snížení ve srovnání s PatchCore. Na referenčním datasetu MVTec dosáhl FastFlow přesnosti 99,4 %.



Obr. 4.2: Model FastFlow (převzato z [39] a upraveno)

4.4 Tvorba datasetu

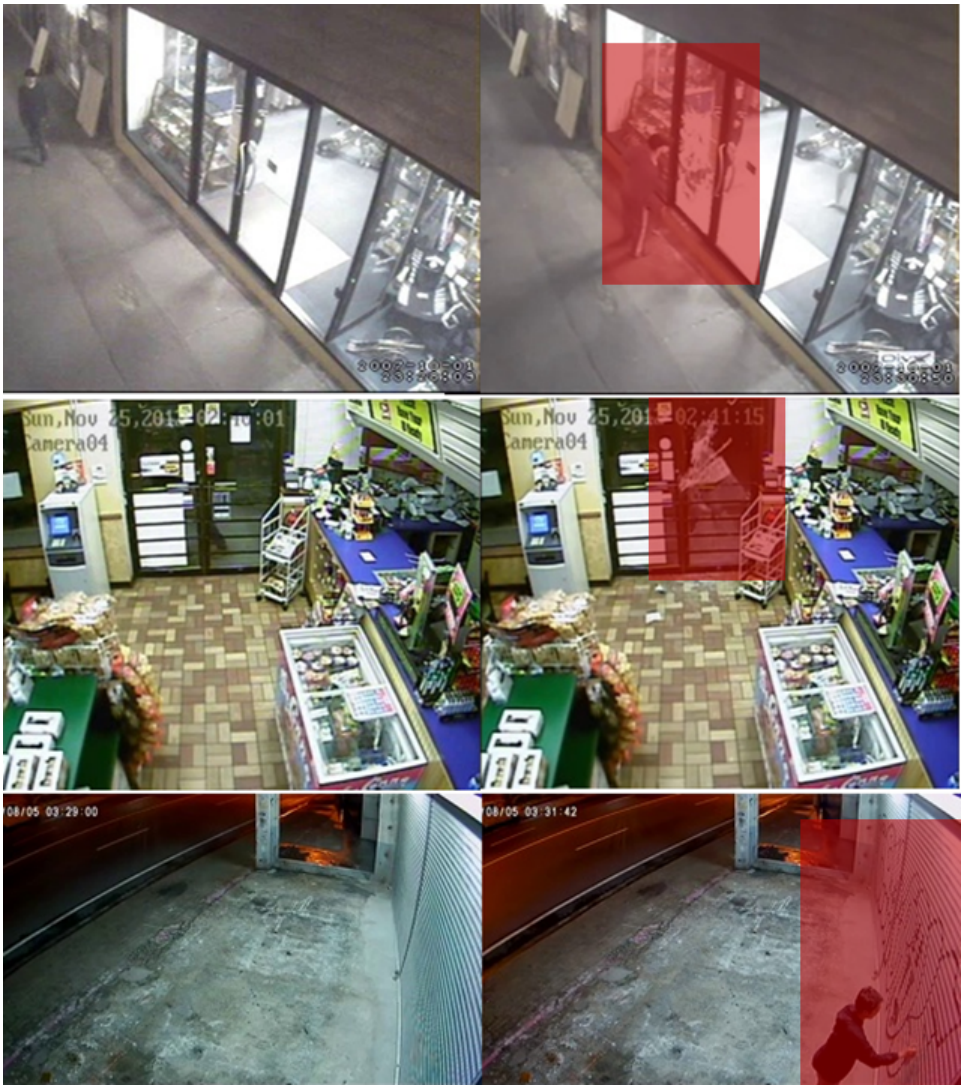
Pro účely trénování a testování modelů neuronových sítí bylo zapotřebí shromáždit dostatečné množství videosekvencí, které co nejlépe reprezentují vybrané typy anomálií. Tato práce je úžeji zaměřena na odhalování vandalismu na objektech vyskytujících se ve videosekvenci, tedy zejména jejich poškození nebo odstranění (krádež). Z tohoto důvodu byl zvolen obsáhlý dataset UCF Crime [33]. Jedná se o soubor videí vytvořený na University of Central Florida v USA, který se používá pro trénování a testování algoritmů pro detekci trestných činů na veřejných prostranstvích. V rámci celkem 128 hodin videa je zde pokryto 13 různých typů anomálií se zaměřením na kriminální činnost. Videa byla získána z veřejně dostupných zdrojů (YouTube, LiveLeak), normalizována na rozlišení 320 × 240 px a snímkovou frekvenci 30 FPS (Frames Per Second).

Dataset obsahuje i anomálie mimo zaměření této práce, jako střelba nebo exploze, vybrána proto byla pouze část videosekvencí z celkem tří kategorií. Podmínkou při výběru videí bylo, aby vyobrazená anomálie byla vždy jasně zřetelná a definovaná. Jak bylo zmíněno, Anomalib v aktuální verzi podporuje datasety pouze na úrovni snímků. Videosekvence byly tedy rozděleny tak, že každou sekundu videa byl zachycen jeden snímek, který byl zařazen do datasetu. V tabulce 4.2 je shrnuto složení trénovací datové sady. Uvedená je zde kategorie UCF Crime, ze které videosekvence pochází, majoritní typ anomálie, který se ve videosekvencích vyskytuje, počet vybraných videí, stopáž a celkový počet snímků.

Tab. 4.2: Složení datasetu pro trénování modelů

UCF Crime	Typ anomálie	Počet videí	Stopáž	Počet snímků
Burglary	Poškozování	13	23,48 min	1215
Robbery	Krádež	9	9,5 min	453
Vandalism	Výtržnictví	15	23,95 min	1216

Na obrázku 4.3 jsou příklady anomálií z videosekvencí v trénovacím datasetu. Vlevo jsou pro porovnání normální snímky a vpravo jsou červeně vyznačeny anomální situace. V prvních dvou ukázkách dochází k rozbití prosklených dveří a v druhém případě navíc vniknutí do objektu a odcizení zboží. Ve třetí ukázce jsou poškozena (posprejována) vrata.



Obr. 4.3: Ukázky anomálií z trénovací datové sady [33]

4.5 Trénování modelů

Pro konfiguraci modelů jsou používány konfigurační soubory YAML, které obsahují informace o konfiguraci modelu, jako jsou hyperparametry, architektura modelu, zdroj trénovacích dat a další nastavení. Tyto soubory slouží k reprezentaci dat ve

strukturované podobě a jejich obsah je snadno čitelný jak pro člověka, tak i pro stroj.

Pro trénování byl použit skript `train.py`. Anomalib na konci každé epochy ověřuje aktuální stav modelu na validačních datech a vyhodnocuje jeho přesnost. V souboru `.yaml` lze pro daný model nastavit rozhodovací hranici přesnosti, při jejímž překročení se trénování ukončí. V opačném případě se pokračuje další epochou. Anomalib poskytuje adaptivní mechanismus, který během validace optimalizuje hodnotu rozhodovací hranice dynamicky na základě metriky Skóre F1. Výstupem fáze trénování je soubor checkpoint (přípona `.ckpt`), ve kterém jsou uloženy parametry natrénovaného modelu jako váhy, biasy, gradienty a také konfigurace neuronové sítě. Ukládání checkpointů umožňuje obnovit naučené parametry a pokračovat v trénování modelu.

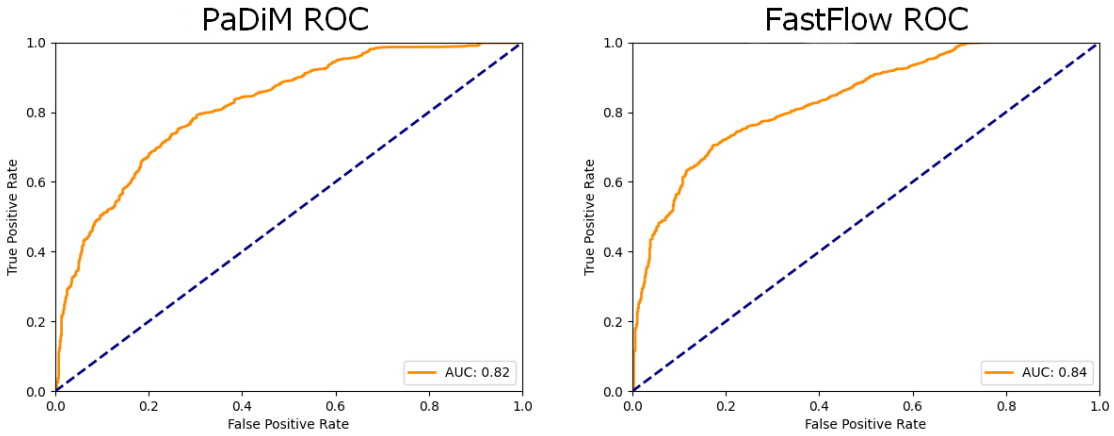
Značná velikost vytvořeného datasetu nedovoluje natrénování modelu na zařízení s nízkým výkonem a malou fyzickou pamětí. Trénování by trvalo příliš dlouho a především by hrozilo zahlcení paměti a tedy okamžité zastavení trénovacího procesu. Z těchto důvodů bylo natrénování vybraných modelů provedeno na výkonném výpočetním stroji s dedikovanou grafickou kartou. Základní specifikace trénovacího počítače jsou shrnuty v tabulce 4.3. Počítač disponuje 12jádrovým procesorem s podporou SMT, díky čemuž je schopen vykonávat až 24 vláken současně. V konfiguračních souborech byl tedy nastaven parametr `num_workers` na hodnotu 24, aby byl při trénování maximálně využit výpočetní potenciál. Grafická karta je schopna paralelně zpracovávat velké množství matematických operací, čímž výrazně urychluje proces trénování modelů neuronových sítí.

Tab. 4.3: Specifikace trénovacího výpočetního stroje

Procesor	AMD Ryzen 9 3900X 12-Core
Paměť RAM	32 GB
Grafická karta	NVIDIA GeForce RTX 2080Ti 11 GB
Paměť SSD	1 TB
Paměť HDD	2 TB
Operační systém	Ubuntu 20.04.5

V konfiguračním soboru se nastavuje poměr, v jakém bude celý dataset rozdělen na trénovací, testovací a validační část. Tento poměr byl nastaven tak, že 80 % dat je použito pro trénování a zbylých 20 % na validaci a testování. Po dokončení trénování je výkon modelu ověřen na testovacích datech. K vyhodnocení Anomalib používá popsané metriky AUROC a Skóre F1. Průběh trénování vybraných modelů

a výsledky evaluačních metrik jsou shrnuty v tabulce 4.4. Na obrázku 4.4 jsou pro oba trénované modely uvedeny křivky ROC.



Obr. 4.4: Křivky ROC

Tab. 4.4: Výsledky trénování modelů

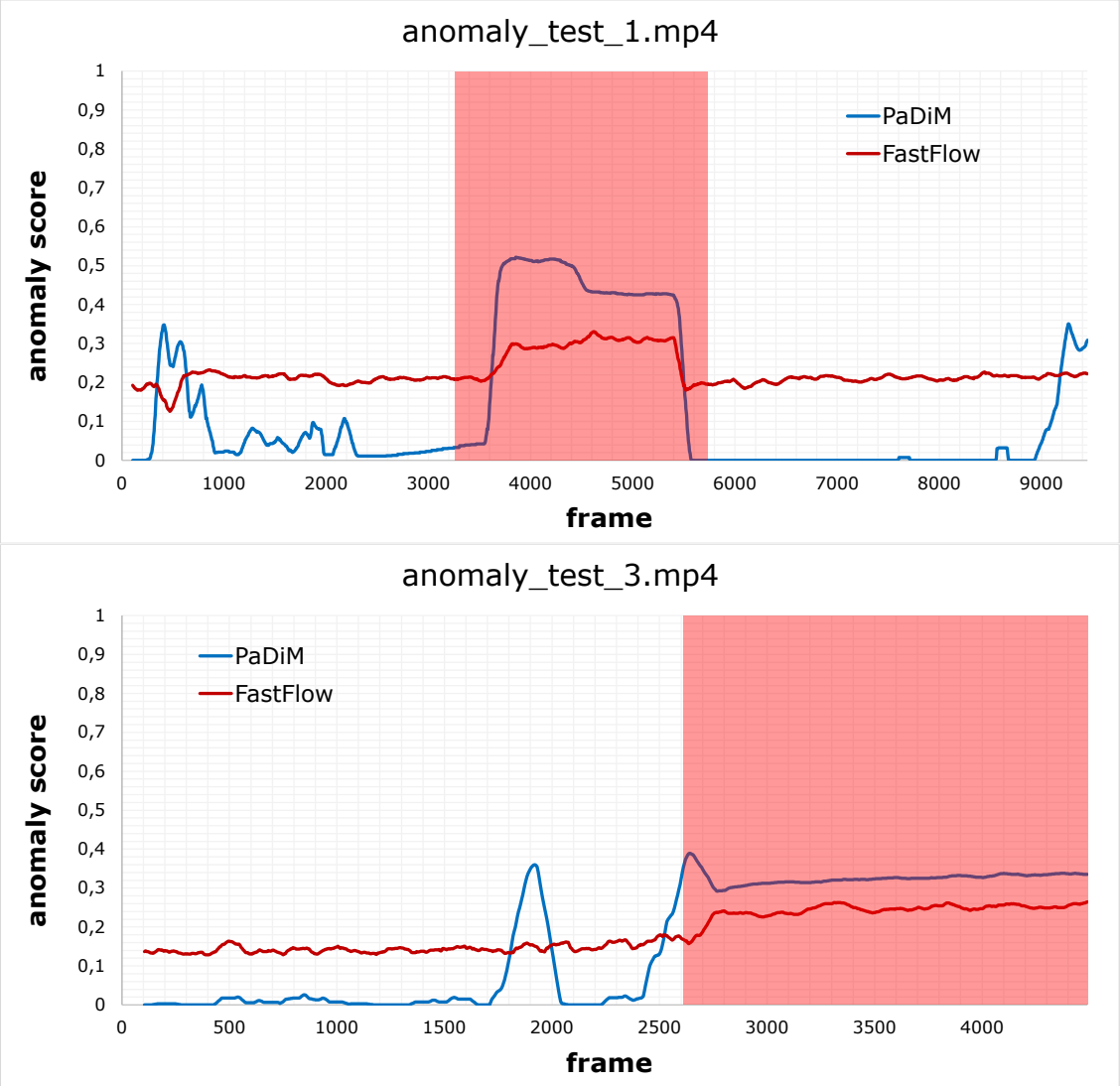
Model	**Doba trénování**	**Propustnost**	**AUROC**	**Skóre F1**
PaDiM	93,31 s	72,89 FPS	0,824	0,910
FastFlow	46,82 s	75,19 FPS	0,836	0,913

4.6 Vyhodnocení anomálnosti

Výstupem modelu je pro každý testovaný snímek indikátor anomaly score, který nabývá hodnot od 0 do 1 a udává míru odchylky jednotlivých snímků od normálního chování a tedy pravděpodobnost výskytu anomálie. Individuální hodnoty pravděpodobnosti nemusí ale vždy přesně vyjadřovat skutečnou míru anomálie. Například některé snímky mohou obsahovat velké odchylky, zatímco jiné mohou být blíže k normálnímu chování. Při uvažování pouhé závislosti na jednotlivých hodnotách pravděpodobnosti by mohly být přehlédnuty nebo naopak nadhodnoceny určité anomálie. Pro zvýšení přesnosti vyhodnocení byly proto hodnoty pravděpodobnosti zprůměrovány pomocí posuvného okna (sliding window). Tato technika průměruje hodnoty z více snímků a vytváří tak plynulý průběh hodnot anomaly score. Tímto způsobem se vyhlazují výkyvy a nepřesnosti způsobené individuálními snímky.

```
def sliding_window(data, window_size):
    window = np.ones(int(window_size))/float(window_size)
    return np.convolve(data, window)
```

Průběh detekce obou testovaných modelů na ukázkových videosekvencích je na obrázku 4.5. Skutečná anomálie se nachází v oblasti, která je naznačena červeným podbarvením. Je zřejmé, že oba modely jsou schopny vyskytující se anomálnost detekovat lokálním zvýšením hodnoty anomaly score, které se ale liší v závislosti na vybraném modelu a testované videosekvenci. Nastavení statické rozhodovací hranice by tedy mohlo vést k tomu, že některé anomální snímky budou nesprávně klasifikovány jako normální nebo naopak.



Obr. 4.5: Průběh predikce modelů na ukázkových videosekvencích

Rozhodovací hranice byla tedy zvolena dynamicky na základě metriky MAD (Median Absolute Deviation). Jedná se o robustní metriku pro měření variability v datech, která je založena na mediánu absolutních odchylek od mediánu dat. Použití této metriky umožňuje dynamické nastavení rozhodovací hranice tak, aby se minimalizoval počet nesprávně klasifikovaných anomálií.

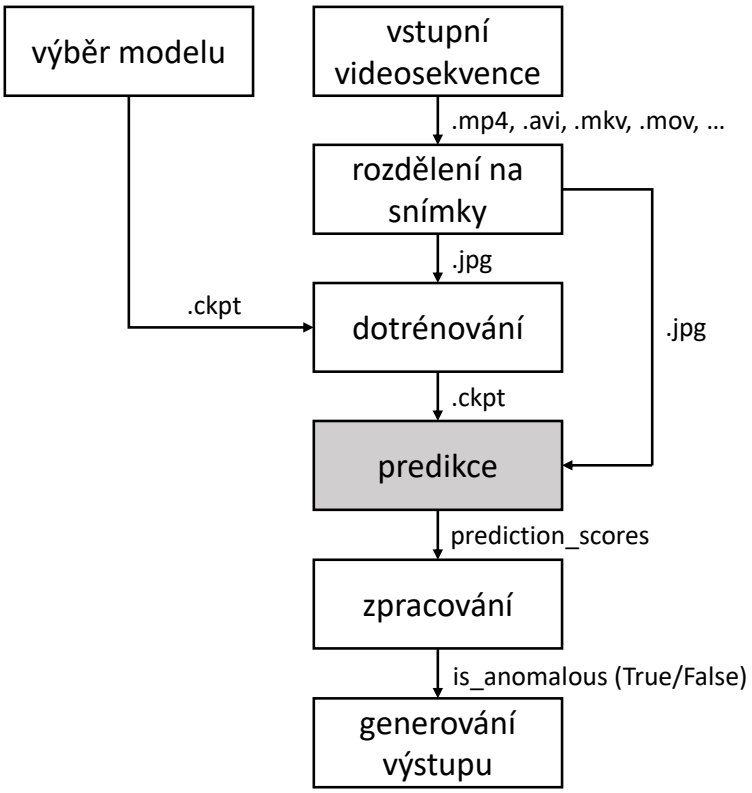
```
median = statistics.median(prediction_scores)
absolute_deviations = []
for score in prediction_scores:
   absolute_deviations.append(abs(score-median))
   mad = statistics.median(absolute_deviations)
```

4.7 Tvorba aplikace

Pro naprogramování aplikace byl zvolen programovací jazyk Python. Jedná se o vysokoúrovňový programovací jazyk, který poskytuje velké množství knihoven a nástrojů pro práci s daty. Tyto nástroje usnadňují práci s velkými soubory dat a umožňují rychlou implementaci algoritmů strojového učení.

Vytvořena byla aplikace, jejíž schéma je naznačeno na obrázku 4.6. Po spuštění hlavního skriptu `main.py` je uživatel vyzván k zadání cesty k vstupní testované videosekvenci a výběru modelu detekce anomálií (PaDiM nebo FastFlow). Videosekvence je rozdělena na snímky, které jsou uloženy do adresáře `temp/frames`. Ve výchozím nastavení se ukládá každý pátý snímek videosekvence. Pro účely dotrénování jsou z úplného začátku videosekvence vybrány dva snímky, které jsou uloženy do složky `temp/retrain` na kterých proběhne dotrénování vybraného modelu za účelem adaptace na novou situaci.

Program nejdříve načte parametry předem natrénovaného modelu, který je uložen v souboru `model.ckpt` v adresáři `/checkpoint/{nazev_modelu}` a provede dotrénování na nových datech. Dotrénovaný model je uložen jako další soubor `model.ckpt` do složky `/temp/{nazev_modelu}`. Následuje inference na snímcích ze vstupní videosekvence. Tato část je z hlediska výkonu i doby trvání nejnáročnější. Výsledné predikce jsou následně zpracovány pomocí posuvného okna a vypočtena je pro ně hodnota MAD, která slouží jako rozhodovací hranice pro detekci anomálií. Po dokončení se vypíšou výsledky detekce do konzolového okna a do domovského adresáře projektu se vygeneruje video `output.mp4`. V něm se kromě čísla aktuálního snímku vyobrazí v případě detekované anomálie nápis „ANOMALY“, jak je vidět na obrázku 4.7.



Obr. 4.6: Návrh schématu aplikace

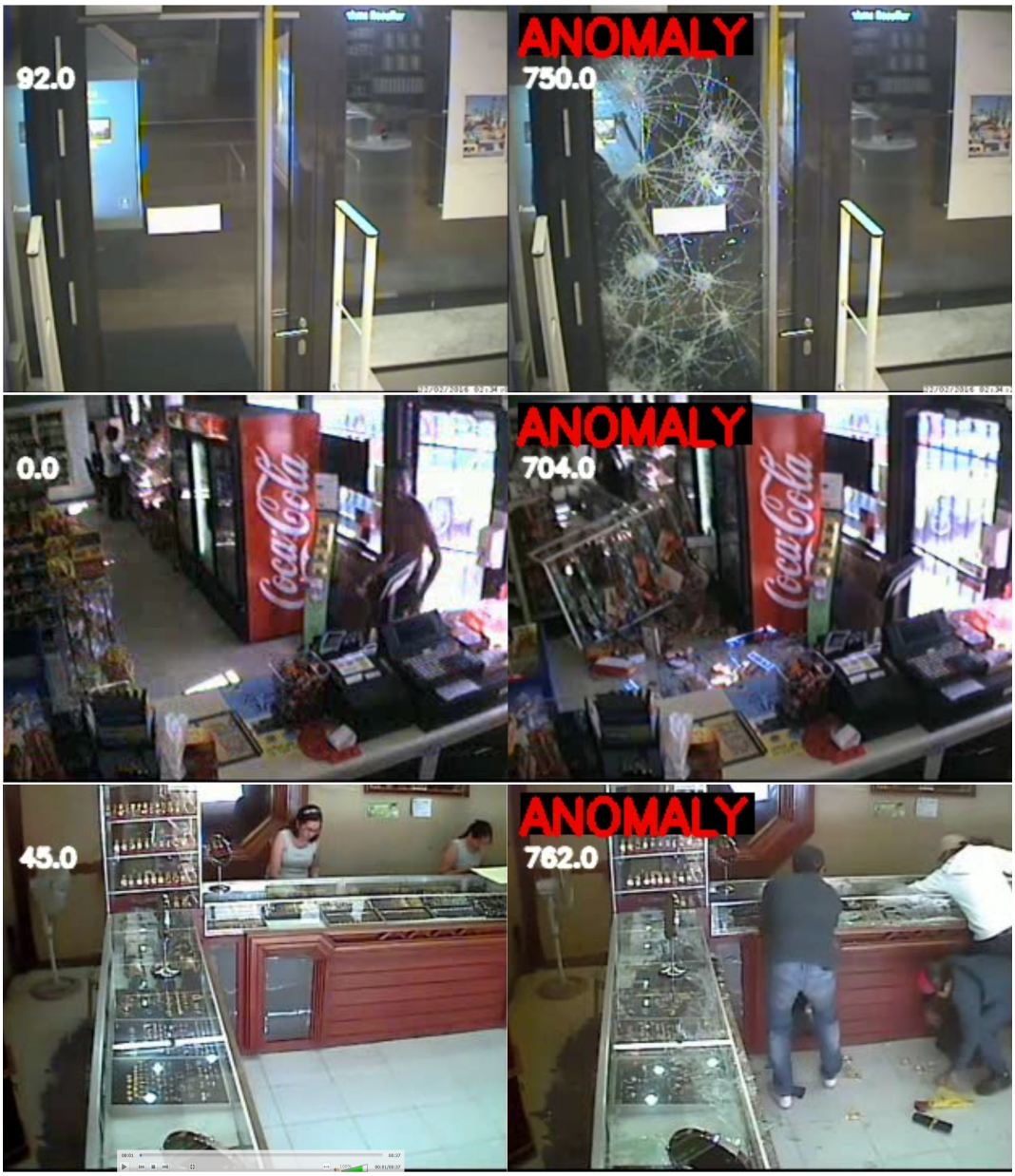
4.8 Testování a výsledky

Testování výsledného algoritmu bylo provedeno na zařízení s nízkým výpočetním výkonem. Jeho specifikace jsou vypsané v tabulce 4.5. Procesor má čtyři fyzická jádra, ale disponuje technologií Intel hyper-threading, která umožňuje jednomu fyzickému jádru procesoru fungovat jako dva nezávislé logické procesory (vlákna). Celkový počet logických procesorů je tedy zdvojnásoben na hodnotu 8. To bylo ověřeno pomocí příkazu `cpu_count()`.

```
In [1]: import multiprocessing
In [2]: multiprocessing.cpu_count()
Out[2]: 8
```

Pro účely testování byl ponížen počet využitých procesorů na čtyři včetně paralelního zpracování pomocí příkazů níže. Grafická karta, kterou disponuje testovací počítač, byla pro testování zakázána a v tabulce je uvedena pouze pro úplnost.

```
physical_cores = multiprocessing.cpu_count()/2
pool = multiprocessing.Pool(processes=physical_cores)
```



Obr. 4.7: Ukázky z výstupních videí

Tab. 4.5: Specifikace testovacího počítače

Procesor	Intel Core i5-8265U 4-Core 1,60 GHz
Paměť RAM	8 GB
Grafická karta	NVIDIA GeForce MX130
Paměť SSD	1 TB
Operační systém	Windows 11

4.8.1 Porovnání modelů

Pro testování a porovnání úspěšnosti modelů detekce anomálií byly tyto modely srovnány na videosekvencích, které obsahují vybrané typy anomálií. Testovací videosekvence jsou součástí přílohy bakalářské práce. Vyhodnocení bylo provedeno na úrovni jednotlivých snímků a rozhodnutí o skutečné anomálnosti bylo učiněno na základě subjektivního vnímání autora práce. Na zvolených videosekvencích jsou anomálie dobře zřetelné a definované, i proto je přesnost obou modelů velmi dobrá, viz Tab. 4.6. Je důležité zmínit, že pokud jsou anomálie hůře viditelné nebo jsou velmi subtilní, úspěšnost detekce a tedy přesnost modelů klesá. Ačkoliv model FastFlow dosahuje na ukázkových videosekvencích lepší přesnosti, testováním na větším počtu videí s různými charaktery anomálií se ukázal jako robustnější model PaDiM.

Tab. 4.6: Porovnání přesnosti modelů na vybraných videosekvencích

videosekvence	délka	**PaDiM**	**FastFlow**
anomaly_test_1.mp4	5,25 min	88,95 %	92,84 %
anomaly_test_2.mp4	1,15 min	87,95 %	93,98 %
anomaly_test_3.mp4	2,48 min	89,70 %	93,45 %

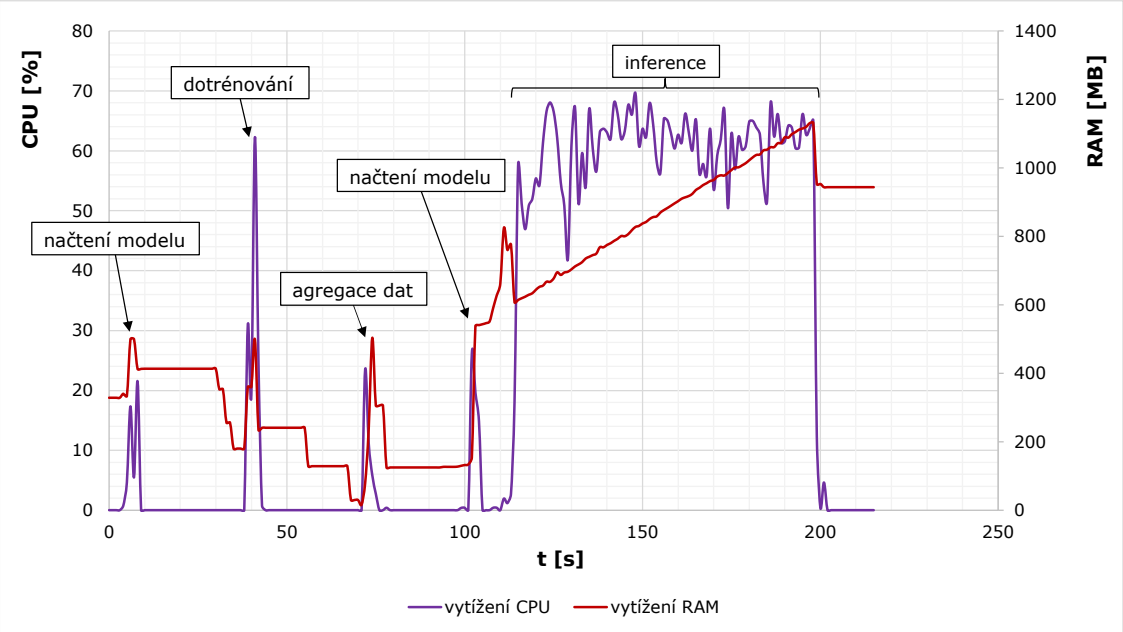
4.8.2 Měření výkonu a vytížení

Pro zajištění dostatečného výkonu na zařízení s nízkými výpočetními kapacitami bylo třeba zvolit metody přizpůsobené pro tyto podmínky a následně provést analýzu vytížení zařízení. Ta zahrnuje zejména měření využití CPU a RAM. Je důležité, aby vytížení testovacího zařízení bylo na přijatelné úrovni za účelem optimálního fungování modelu. Vysoké vytížení zařízení může vést ke zpomalení výkonu a zhoršení kvality výsledků detekce anomálií. Pro měření výkonu je při zpracování obrazových dat klíčová hodnota FPS (Frames Per Second). Ta udává počet snímků, který je model schopen zpracovat za sekundu. Pro rychlou a efektivní detekci anomálií ve videosekvencích je důležitá dostatečně vysoká hodnota FPS, zejména pak v real-time nasazeních.

Aby bylo zjištěno, jak jsou procesor a paměť vytíženy v průběhu vykonávání algoritmu, bylo provedeno měření na odpovídajícím PID pomocí vlastního skriptu `usage.py`. PID (Process Identifier) je číselný identifikátor přiřazený každému procesu v operačním systému.

```
cpu_percent = process.cpu_percent()
mem_usage = process.memory_info().rss/1024/1024 # MB
```

Průběh vytížení CPU v procentech (%) a RAM v megabajtech (MB) při vykonávání algoritmu je vyobrazen graficky na obrázku 4.8. Vyznačeny a popsány jsou zde významné události, které znamenají okamžitý nárůst vytížení. Konkrétně se jedná o načtení a inicializaci předem natrénovaného modelu, dotrénování, agregace nově získaných dat. Testované video je dlouhé 1,15 min s celkovým počtem 414 snímků určených ke klasifikaci.



Obr. 4.8: Vytížení procesoru a paměti

Z grafu je zřejmé, že k největšímu soustavnému vytížení procesoru testovacího počítače dochází při inferenci. Toto vytížení však nepřesahuje hranici 70 %, což je pro většinu zařízení s nízkým výpočetním výkonem akceptovatelné. Při inferenci se do paměti načítají testovaná data a RAM se tak v čase postupně zaplňuje. To může být problém při detekci anomálií u delších videí. Řešením je nastavit počet vybraných snímků z videosekvence dynamicky v závislosti na její délce. Na jednu stranu se sníží plynulost detekce, ale nedojde k vyčerpání paměťových prostředků. Dalším řešením může být snížení rozlišení testovaných snímků, což ale může vést ke zhoršení kvality detekce.

Při detekci anomálií je žádoucí dosáhnout co nejvyšší hodnoty FPS, aby bylo možné anomálie detekovat ideálně v reálném čase. Měření ukázalo, že PaDiM dosahuje při inferenci na testovacím zařízení s nízkým výpočetním výkonem průměrně **6,09 FPS** a FastFlow průměrně **4,52 FPS**. Za předpokladu, že se anomálie vyskytují jen velmi zřídka, není zapotřebí provádět detekci na každém snímku videosekvence a stačí analyzovat pouze vybrané snímky s určitým časovým rozestupem.

Například u videosekvencí o snímkové frekvenci 30 FPS stačí při použití zvolených algoritmů detekovat každý 15. - 20. snímek, aby bylo dosaženo real-time detekce anomálií. Z tohoto důvodu jsou naměřená FPS dostačující pro nasazení modelů v reálném čase.

Závěr

Tato práce se zabývá problematikou detekce anomálií ve videosekvencích na zařízeních s nízkým výpočetním výkonem. První kapitola práce se zaměřuje na přehled strojového učení, kde jsou popsány typy strojového učení a tradiční metody využívané při detekci anomálií. Druhá kapitola se soustředí na popis umělých neuronových sítí, které jsou v současnosti stále více využívány v oblasti detekce anomálií. Detailněji jsou popsány FNN a CNN jakožto klíčové architektury v oblasti počítačového vidění a detekce anomálií. Ve třetí kapitole je popsána problematika detekce anomálií ve videosekvencích a vybrané metody, které patří v současnosti k nejpoužívanějším v této oblasti.

Detekce je dosaženo pomocí vybraných DL modelů. Pro účely jejich natrénování byly shromážděny videosekvence obsahující vybrané typy anomálií. Modely byly pomocí tohoto datasetu natrénovány na výkonném výpočetním stroji. Před klasifikací na zařízení s nízkým výpočetním výkonem proběhne dotrénování předem natrénovaného modelu na několika snímcích z testované videosekvence. Pro implementaci modelů neuronových sítí je použita knihovna Anomalib. Byla vytvořena aplikace, která je na předloženém videu schopna provést detekci anomálií a poskytnout uživateli výsledky detekce v textové podobě i ve formě videa.

Experimenty na reálných videosekvencích prokázaly, že použitá metoda je schopná efektivně detekovat anomálie s dobrou přesností, zatímco zůstává dostatečně rychlá a robustní pro použití na zařízeních s omezenými výpočetními zdroji. S učitými modifikacemi algoritmu je metoda použitelná i pro nasazení v reálném čase.

Literatura

[1] Nassif, A., Talib, M., Nasir, Q. & Dakalbab, F. Machine learning for anomaly detection: A systematic review. 2021. Dostupné z: doi:10.1109/ACCESS.2021.3083060

[2] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L. & Lopez, A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*. 2020, 189-215. Dostupné z: doi:10.1016/j.neucom.2019.10.118

[3] Doshi, K. & Yilmaz, Y. A modular and unified framework for detecting and localizing video anomalies. *Proceedings Of The IEEE/CVF Winter Conference On Applications Of Computer Vision*. 2022, 3982-3991. Dostupné z: doi:10.48550/arXiv.2103.11299

[4] Maqsood, R., Bajwa, U., Saleem, G., Raza, R. & Anwar, M. Anomaly recognition from surveillance videos using 3D convolution neural network. *Multimedia Tools And Applications*. 2021, 18693-18716. Dostupné z: doi:10.1007/s11042-021-10570-3

[5] Smeureanu, S., Ionescu, R., Popescu, M. & Alexe, B. Deep appearance features for abnormal behavior detection in video. *International Conference On Image Analysis And Processing*. 2017, 779-789. Dostupné z: doi:10.1007/978-3-319-68548-9_70

[6] Chen, K. Deep and modular neural networks. *Springer Handbook Of Computational Intelligence*. 2015, 473-494. Dostupné z: doi:10.1007/978-3-662-43505-2_28

[7] Nayak, R., Pati, U. & Das, S. A comprehensive review on deep learning-based methods for video anomaly detection. *Image And Vision Computing*. 2021. Dostupné z: doi:10.1016/j.imavis.2020.104078

[8] Laxhammar, R. Anomaly detection in trajectory data for surveillance applications. 2011. Dostupné z: urn:nbn:se:oru:diva-17235

[9] Xu, K., Jiang, X. & Sun, T. Anomaly detection based on stacked sparse coding with intraframe classification strategy. *IEEE Transactions On Multimedia*. 2018, 1062-1074. Dostupné z: doi:10.1109/TMM.2018.2818942

[10] Kiran, B., Thomas, D. & Parakkal, R. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal Of Imaging*. 2018. Dostupné z: doi:10.3390/jimaging4020036

[11] Oord, T. Machine learning classification tool for innovation projects. [online]. 2019 [cit. 2023-05-16]. Dostupné z URL: http://essay.utwente.nl/78574/

[12] Osiński, B. What is reinforcement learning? The Complete Guide. [online]. 2022 [cit. 2023-04-12]. Dostupné z URL: https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/

[13] Ray, S. A quick review of machine learning algorithms. *2019 International Conference On Machine Learning, Big Data, Cloud And Parallel Computing (COMITCon)*. 2019, 35-39. Dostupné z: doi:10.1109/COMITCon.2019.8862451

[14] Weiss, K., Khoshgoftaar, T. & Wang, D. A survey of transfer learning. *Journal Of Big Data*. 2016, 1-40. Dostupné z: doi:10.1186/s40537-016-0043-6

[15] Bengio, Y., Courville, A. & Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions On Pattern Analysis And Machine Intelligence*. 2013, 1798-1828. Dostupné z: doi:10.1109/TPAMI.2013.50

[16] Wang, Y., Yao, Q., Kwok, J. & Ni, L. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (csur)*. 2020. Dostupné z: doi:10.1145/3386252

[17] Zhao, M., Chen, J. & Li, Y. A Review of Anomaly Detection Techniques Based on Nearest Neighbor. *2018 International Conference On Computer Modeling, Simulation And Algorithm (CMSA 2018)*. 2018, 290-292. Dostupné z: doi:10.2991/cmsa-18.2018.65

[18] Agrawal, S. & Agrawal, J. Survey on anomaly detection using data mining techniques. *Procedia Computer Science*. 2015, 708-713. Dostupné z: doi:doi.org/10.1016/j.procs.2015.08.220

[19] Suzuki, K. Artificial neural networks: methodological advances and biomedical applications. 2011. ISBN 978-9533072432

[20] Svozil, D., Kvasnicka, V. & Pospichal, J. Introduction to multi-layer feed-forward neural networks. *Chemometrics And Intelligent Laboratory Systems*. 1997, 43-62. Dostupné z: doi:10.1016/S0169-7439(97)00061-0

[21] Medsker, L. & Jain, L. Recurrent neural networks. *Design And Applications*. 2001, 64-67. ISBN 978-0849371813

[22] Javaid Nabi Recurrent Neural Networks (RNNs). [online]. 2019 [cit. 2023-05-06]. Dostupné z URL: https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85

[23] Abid Ali Awan What are Recurrent Neural Networks (RNN). [online]. 2022 [cit. 2023-05-06]. Dostupné z URL: `https://www.datacamp.com/tutorial/tutorial-for-recurrent-neural-network`

[24] Albawi, S., Mohammed, T. & Al-Zawi, S. Understanding of a convolutional neural network. *2017 International Conference On Engineering And Technology (ICET)*. 2017, 1-6. Dostupné z: doi:10.1109/ICEngTechnol.2017.8308186

[25] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. & Others Recent advances in convolutional neural networks. *Pattern Recognition*. 2018, 354-377. Dostupné z: doi:10.1016/j.patcog.2017.10.013

[26] Li, Z., Liu, F., Yang, W., Peng, S. & Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions On Neural Networks And Learning Systems*. 2021, 1-21. Dostupné z: doi:10.1109/TNNLS.2021.3084827

[27] Goodfellow, I., Bengio, Y. & Courville, A. Convolutional networks. *Deep Learning*. 2016, 330-372. ISBN 978-0262035613

[28] Chang, J. & Sha, J. An efficient implementation of 2D convolution in CNN. *IEICE Electronics Express*. 2017. Dostupné z: doi:10.1587/elex.13.20161134

[29] Khan, S., Hafeez, Q., Khalid, M., Alroobaea, R., Hussain, S., Iqbal, J., Almotiri, J. & Ullah, S. Anomaly detection in traffic surveillance videos using deep learning. *Sensors*. 2022. Dostupné z: doi:10.3390/s22176563

[30] Profentzas, C., Almgren, M. & Landsiedel, O. Performance of deep neural networks on low-power IoT devices. *Proceedings Of The Workshop On Benchmarking Cyber-Physical Systems And Internet Of Things*. 2021, 32-37. Dostupné z: doi:10.1145/3458473.3458823

[31] Sliwa, B., Piatkowski, N. & Wietfeld, C. LIMITS: Lightweight machine learning for IoT systems with resource limitations. *ICC 2020-2020 IEEE International Conference On Communications (ICC)*. 2020, 1-7. Dostupné z: doi:10.48550/arXiv.2001.10189

[32] Lane, N., Bhattacharya, S., Georgiev, P., Forlivesi, C., Jiao, L., Qendro, L. & Kawsar, F. Deepx: A software accelerator for low-power deep learning inference on mobile devices. *2016 15th ACM/IEEE International Conference On Information Processing In Sensor Networks (IPSN)*. 2016, 1-12. Dostupné z: doi:10.1109/IPSN.2016.7460664

[33] Sultani, W., Chen, C. & Shah, M. Real-world anomaly detection in surveillance videos. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*. 2018, 6479-6488. Dostupné z: doi:10.48550/arXiv.1801.04264

[34] Zhang, X., Li, X., Feng, Y. & Liu, Z. The use of ROC and AUC in the validation of objective image fusion evaluation metrics. *Signal Processing*. 2015, 38-48. Dostupné z: doi:10.1016/j.sigpro.2015.03.007

[35] Lai, Y. A comparison of traditional machine learning and deep learning in image recognition. *Journal Of Physics: Conference Series*. 2019. Dostupné z: doi:10.1088/1742-6596/1314/1/012148

[36] Deep Learning for Anomaly Detection. [online]. 2020 [cit. 2023-05-24]. Dostupné z URL: `https://ff12.fastforwardlabs.com/`

[37] Akcay, S., Ameln, D., Vaidya, A., Lakshmanan, B., Ahuja, N. & Genc, U. Anomalib: A Deep Learning Library for Anomaly Detection. 2022. Dostupné z: doi:10.1109/ICIP46576.2022.9897283

[38] Defard, T., Setkov, A., Loesch, A. & Audigier, R. Padim: a patch distribution modeling framework for anomaly detection and localization. *International Conference On Pattern Recognition*. 2021, 475-489. Dostupné z: doi:10.1007/978-3-030-68799-1_35

[39] Yu, J., Zheng, Y., Wang, X., Li, W., Wu, Y., Zhao, R. & Wu, L. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. 2021. Dostupné z: doi:10.48550/arXiv.2111.07677

Seznam symbolů a zkratek

GPU	Graphics Processing Unit – grafický procesor
ML	Machine Learning – strojové učení
SVM	Support Vector Machine – metoda podpůrných vektorů
k-NN	k-Nearest Neighbor – algoritmus k-nejbližších sousedů
TL	Transfer Learning – přenos učení
FSL	Few-Shot Learning – metoda učení pomocí malého množství vzorků
ANN	Artificial Neural Network – umělá neuronová síť
FNN	Feed-forward Neural Network – dopředná neuronová síť
ReLU	Rectified Linear Unit – usměrněná lineární funkce
DeepAE	Deep Auto Encoder – hluboký autoenkodér
RNN	Recurrent Neural Network – rekurentní neuronová síť
BPTT	Back Propagation Through Time – zpětná propagace v čase
LSTM	Long Short Term Memory – typ rekurentní neuronové sítě
CNN	Convolutional Neural Network – konvoluční neuronová síť
VCA	Video Content Analysis – analýza obsahu videa
DNN	Deep Neural Network – hluboká neuronová síť
CPU	Central Processing Unit – centrální procesorová jednotka
PCA	Principal Component Analysis – analýza hlavních komponent
OCC	One Class Classification – jednoduchá klasifikace
TP	True Positive – skutečně pozitivní
TN	True Negative – skutečně negativní
FP	False Positive – falešně pozitivní
FN	False Negative – falešně negativní
ROC	Receiver Operating Characteristic – charakteristika klasifikátoru

TPR True Positive Rate – míra pravdivé pozitivity

FPR False Positive Rate – míra falešné pozitivity

AUROC Area Under ROC – plocha pod křivkou ROC

SMT Simultaneous Multi-Threading – současné vícevláknové zpracování

TPU Tensor Processing Unit – procesor pro tenzorové zpracování

RAM Random Access Memory – paměť s libovolným přístupem

PDF Probability Density Function – hustota rozdělení pravděpodobnosti

FPS Frames Per Second – snímky za sekundu

MAD Median Absolute Deviation – medián absolutní odchylky

PID Process Identifier – identifikátor procesu

A Obsah elektronické přílohy

Součástí elektronické přílohy jsou kromě samotné aplikace testovací videosekvence, natrénované modely, konfigurační soubory, skript pro měření vytížení a skript pro trénování modelů.