



**Ekonomická
fakulta**
Faculty
of Economics

**Jihočeská univerzita
v Českých Budějovicích**
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích

Ekonomická fakulta

Katedra aplikované matematiky a informatiky

Diplomová práce

**Segmentace snímků z termokamery aplikací
vybraných konvolučních neuronových sítí**

Vypracoval: Bc. Lukáš Beneda

Vedoucí práce: doc. RNDr. Petr Bartoš, Ph.D.

České Budějovice 2023

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval/a samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to – v nezkrácené podobě/v úpravě vzniklé vypuštěním vyznačených částí archivovaných Ekonomickou fakultou – elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

Datum

Podpis studenta

Poděkování

Rád bych poděkoval Mgr. Ing. Romanu Bumbálkovi za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování mé diplomové práce. Dále bych chtěl také poděkovat doc. RNDr. Petru Bartošovi, Ph.D. za vedení práce, nabídku atraktivního tématu a za pomoc s administrativní úkony spojenými se zpracováváním tématu na Fakultě zemědělské a technologické.

Obsah

PROHLÁŠENÍ.....	3
PODĚKOVÁNÍ	5
OBSAH.....	1
1. ÚVOD.....	4
2. TERMOGRAFIE	6
2.1. ZÁKLADNÍ PRINCIPY TECHNOLOGIE	6
2.2. TERMOKAMERY	6
2.3. OBLASTI VYUŽITÍ TERMOGRAFIE.....	7
3. NEURONOVÉ SÍTĚ A JEJICH APLIKACE	9
3.1. KLASIFIKACE OBRAZU	9
3.2. DETEKCE OBJEKTŮ V OBRAZE.....	10
3.3. KONVOLUČNÍ NEURONOVÉ SÍTĚ.....	10
3.3.1. Vrstvy v konvolučních neuronových sítích	10
3.3.2. Ztrátové funkce	13
3.3.3. Metody optimalizace parametrů neuronové sítě.....	13
3.3.4. Metody vyhodnocování výsledků.....	14
3.3.5. Známé modely konvolučních neuronových sítí.....	15
3.4. PŘÍKLADY APLIKACÍ VE VYBRANÝCH OBORECH.....	16
3.4.1. Aplikace v dopravě	16
3.4.2. Průmyslové aplikace	17
3.4.3. Aplikace v lékařství	17
3.4.4. Zemědělské aplikace.....	17
4. SEGMENTACE OBRAZU	18
4.1. KONVENČNÍ METODY SEGMENTACE OBRAZU	18
4.1.1. Treshold-based algoritmy	18
4.1.2. Edge-based algoritmy	18
4.1.3. Cluster-based algoritmy	18
4.2. NEURONOVÉ SÍTĚ PRO SEGMENTACI	19
4.3. TYPY SEGMENTAČNÍCH ÚKOLŮ	19

4.3.1.	Sémantická segmentace	19
4.3.2.	Segmentace instancí.....	20
4.3.3.	Panoptická segmentace	21
4.4.	SEGMENTACE INSTANCÍ ZA POMOCI NEURONOVÝCH SÍTÍ	21
4.4.1.	Mask R-CNN	21
4.4.2.	YOLACT	22
4.4.3.	TensorMask	22
4.4.4.	YOLOv7	23
4.4.5.	EVA	23
4.4.6.	Mask DINO.....	24
5.	PŘÍPRAVA DATASETU.....	26
5.1.1.	Computer Vision Annotation Tool	26
5.1.2.	Formát datasetu.....	26
5.1.3.	Obsah datasetu	26
5.1.4.	Tvorba anotací datasetu	27
6.	UČENÍ NEURONOVÝCH SÍTÍ.....	29
6.1.1.	Mask R-CNN	29
6.1.2.	YOLOv7	31
6.1.3.	Mask DINO.....	31
7.	VÝSLEDKY TRÉNOVÁNÍ VYBRANÝCH SÍTÍ.....	35
7.1.	VIZUALIZACE VÝSLEDKŮ SÍTÍ	35
7.1.1.	Mask R-CNN	35
7.1.2.	YOLOv7	37
7.1.3.	Mask DINO.....	41
7.2.	POROVNÁNÍ VÝSLEDKŮ	42
8.	ZÁVĚR.....	44
9.	SUMMARY AND KEYWORDS	46
9.1.	SUMMARY.....	46
9.1.1.	CZ	46
9.1.2.	EN	46
9.2.	KEYWORDS.....	46

9.2.1. CZ	46
9.2.2. EN	46
10. SEZNAM POUŽITÝCH ZDROJŮ	47
11. SEZNAM OBRÁZKŮ A TABULEK	56
12. SEZNAM PŘÍLOH.....	58
13. PŘÍLOHY	59

1. Úvod

V posledních několika letech je ve veřejném prostoru hojně skloňované téma umělé inteligence. Je to technologie, která při správném využití může velmi pozitivně ovlivnit náš život. Zároveň je zde ale riziko negativních dopadů na společnost.

Strojové učení je jednou z oblastí umělé inteligence, která přináší možnost automatického osvojení konceptů a znalostí bez nutnosti explicitního naprogramování těchto poznatků. Klasické strojové učení je v posledních letech překonáváno technologií deep learningu. Deep learning využívá technologie neuronových sítí a pokročilých učících algoritmů. Proces přípravy této technologie pro konkrétní úlohu se skládá ze dvou částí, kdy v první z nich probíhá samotné učení na předpřipravených datech a v druhé ověření efektivity a přesnosti pomocí nových, pro neuronovou síť neznámých dat. (Dargan, Kumar, Ayyagari, & Kumar, 2020) (Alzubaidi et al., 2021) (Dong, Wang, & Abbas, 2021)

V dnešní době je v komunitě zabývající se neuronovými sítěmi řešena dlouhá řada výzev od detekce a rozpoznávání dopravních značek (Zhang, Wang, Song, Yan, & Qi, 2022), přes rozlišování jednotlivých kamenů v sypkých materiálech (Yang et al., 2021), rozpoznávání obličejů (Sanchez-Moreno et al., 2021), různé detekce poškození (Pham, Pham, & Dang, 2020) (Wen, Huang, & Guo, 2021), počítání objektů na leteckých snímcích (Wu et al., 2021) (Osco et al., 2021), až po různé výzvy v medicíně (Li, Du, Luo, Wang, & Ding, 2023) (Brodzicki, Jaworek-Korjakowska, Kleczek, Garland, & Bogyo, 2020) (Long, Peng, Song, Xia, & Sang, 2021) (A. Al-Falluji, Dalaf Katheeth, & Alathari, 2021) (Zaridis et al., 2023) nebo právě v zemědělství (Santos, de Souza, dos Santos, & Avila, 2020) (Sarwar, Griffin, Rehman, & Pasang, 2021) (Maksimovich, Fedorov, & Kalichkin, 2023).

Tato diplomová práce se zabývá využitím technologie neuronových sítí pro detekci paznehtu skotu, zejména u krav, v termografických snímcích pro následné automatické vyhodnocení anomálií a případné detekce onemocnění paznehtu, které mohou mít negativní vliv na zdraví zvířete nebo na délku jeho života.

Cíl této práce je připravit a porovnat několik modelů neuronových sítí určených k segmentaci instancí. První část se zabývá samotnou problematikou termografických snímků a aplikace této technologie v praxi. V další kapitole je shrnuta problematika neuronových sítí, a především konvoluční neuronové sítě a jejich aplikace. Ve čtvrté

kapitole jsou zmapovány možnosti řešení pro úlohu segmentace obrazu, včetně segmentace obrazu pomocí neuronových sítí. Druhá polovina práce se zabývá přípravou datasetu pro učení sítí na rozpoznávání kravského paznehtu. Následně je popsáno učení vybraných modelů neuronových sítí, a nakonec jsou jednotlivé vybrané modely porovnány z hlediska úspěšnosti, kvality nebo rychlosti inferencí.

2. Termografie

2.1. Základní principy technologie

Termografie je bezkontaktní metoda snímání tepla na povrchu těles, která se využívá k detekci tepla v lidském těle, v těle živočichů nebo v průmyslu. Termografie využívá infračervené záření, které je vyzařováno tělesy v závislosti na jejich teplotě. Toto infračervené záření je snímáno pomocí speciální kamer, které intenzitu vyzařování převádějí na barvu pro snazší interpretaci snímku. Kamery obsahují obrazový senzor podobný tomu, který je obsažen v běžném fotoaparátu, s tím rozdílem, že je citlivý na jiné vlnové délky. Pro infračervené snímky se používá termín termogram.

Princip, se kterým termokamery pracují, poprvé popsal německý fyzik Max Karl Ernst Ludwig Planck. Jedná se o Planckův vyzařovací zákon, který vyjadřuje závislost intenzity záření absolutně černého tělesa na frekvenci ω . Zákon říká, že těleso vyzařuje energii nespojitě po kvantech a pro každou vlnovou délku vyjadřuje spektrální hustotu intenzity vyzařování, která je definována jako množství energie připadající na jednotkový interval vlnové délky.

2.2. Termokamery

Termokamery umožňují zobrazení infračerveného záření tělesa převedením tohoto záření do viditelného spektra. Pro takto pracující kamery se často užívá i termín termovize a podle toho se používá i pro termogram termín termovizní snímek.

Konstrukce termokamer je obdobná jako u klasického fotoaparátu nebo kamery. Objektiv promítá záření na obrazový senzor, který měří intenzitu tohoto záření. Objektiv velmi často obsahuje infračervený filtr, který propustí na obrazový senzor pouze infračervené záření. Informace z obrazového senzoru je převedena do digitální formy a pomocí dříve popsaného principu vizualizována.

Obrazové senzory, detektory, užívané v termokamerách dělíme obvykle na 2 typy, tepelné a fotonové. Tepelné pracují na principu změny elektrických vlastností materiálu při dopadu infračerveného záření, často jde o odpor nebo kapacitu. Fotonové detektory zjednodušeně počítají dopadající fotony, kvanta energie, jsou tedy citlivější a přesnější, ale při tom se zahřívají. Taková kamera je tedy těžší a dražší.

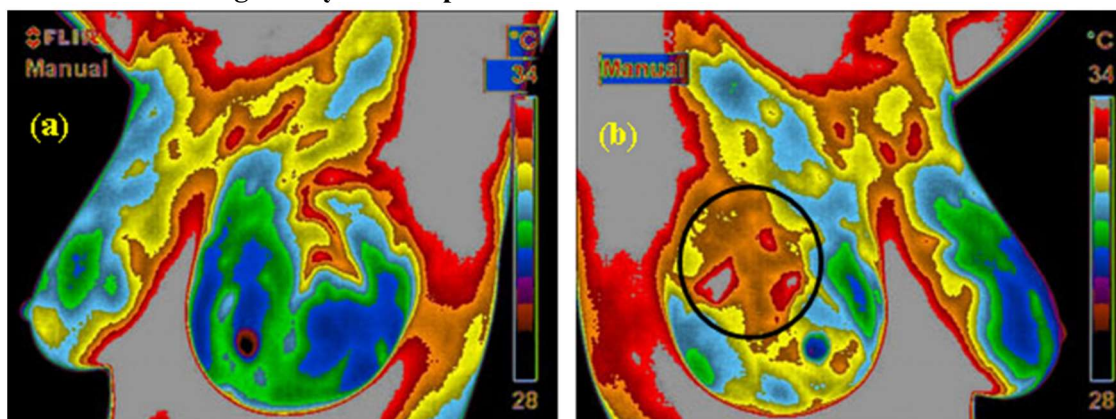
Specifickým typem termokamer jsou radiometrické termokamery, které umožňují určit teplotu v konkrétních bodech termogramu. Není to však zdaleka tak jednoduché, jak by se mohlo zdát, jelikož zde vstupují do procesu další parametry. Hlavním parametrem, který výrazně ovlivňuje výsledek je emisivita tělesa. Absolutně černé těleso má emisivitu 1, lidské tělo 0,96, lesklá tělesa se často blíží až k 0,1. Malá emisivita vede k nižší přesnosti měření.

2.3. Oblasti využití termografie

Termografie má velmi široké pole využití. V praxi je využívána například ve stavebnictví, průmyslu, zemědělství nebo v medicíně. Důležitou roli také hraje při kontrolách a revizích různých zařízení, strojů nebo ochraně životního prostředí.

V medicíně se využívá termografických snímků k detekci změn teploty v těle, které mohou naznačovat onemocnění, například zánět nebo nádor (Jones, 1998). Z praktických příkladů stojí za zmínku identifikace pacientů s revmatoidní artritidou (Wasilewska, Pauk, & Ihnatouski, 2018) nebo detekci nádorů prsou (Silva et al., 2016). Termografie je ovlivňována ve spojení s lidským tělem mnoha faktory, jejich shrnutí uvádí (Fernández-Cuevas et al., 2015), kde jedním z faktorů je například i váha jedince. Další aplikace jsou pak popsány v oblastech diagnostiky jater, dutiny ústní, dermatologii, gynekologii nebo při monitorování krevního tlaku (Lahiri, Bagavathiappan, Jayakumar, & Philip, 2012).

Obrázek 1: Termografický snímek prsu



Zdroj: Lahiri, Bagavathiappan, Jayakumar, & Philip, 2012

Termografie v zemědělství je u nás známa především jako nástroj pro detekci zvířat z bezpilotních letounů před sečením, tato činnost je většinou zajišťována myslivci nebo neziskovými spolky. Ve světě se příkladů najde mnoho od sledování ovoce a následného plánování zavlažování (Li et al., 2019), přes kontrolu kvality extra panenského olivového

oleje (Izquierdo et al., 2020), detekce plevelu (Fathipoor, Shah-Hosseini, & Arefi, 2023) až po sledování zdraví kuřat (Xiong et al., 2019).

Ve stavebnictví je termografie využívána především pro detekci tepelných mostů budov nebo při zateplování (Asdrubali et al., 2018). Další aplikací je odhalování namáhaných míst na zemnicích kotvách (Min, Jang, & Yoon, 2023) nebo testování a průzkumech (Gong, Peng, He, & Wang, 2015). V průmyslu je využívána hlavně při výrobních procesech zejména se využívá ke zjišťování poruch (Djavadifar, Graham-Knight, Körber, Lasserre, & Najjaran, 2022) (Sampath et al., 2023) (Cao, Wang, Qi, Wang, & Jiang, 2019), ve vývoji pomáhá odhalovat nevhodné použití materiálu nebo nepředvídané úniky.

Obrázek 2: Termografický snímek domu



Zdroj: Bagavathiappan, Lahiri, Saravanan, Philip, & Jayakumar, 2013

Při revizích je využívána především pro kontrolu stavu součástí strojů nebo zařízení. Z praktických často vídaných příkladů jsou zmiňovány například kontroly ložisek větrných elektráren, spojovacího materiálu na fotovoltaických elektrárnách, kontroly dalších elektrických součástí nebo inspekce jiných mechanických částí. Termovize je také využívána v jaderné energetice nebo při kontrolách svárů. (Bagavathiappan, Lahiri, Saravanan, Philip, & Jayakumar, 2013) (Sreeshan, Dinesh, & Renji, 2020) (Waqar Akram et al., 2019)

3. Neuronové sítě a jejich aplikace

Neuronové sítě byly poprvé teoreticky popsány v roce 1943 neurofyziologem Warrenem McCullochem a matematikem W. S. Pittsem v práci "A Logical Calculus of the Ideas Immanent in Nervous Activity". Tato práce byla považována za první teoretický popis neuronové sítě jako počítačového modelu pro simulaci fungování lidského mozku. (McCulloch & Pitts, 1943)

V následujících desetiletích byly vytvořeny první konkrétní modely neuronových sítí, jako například perceptron, který byl vyvinut v roce 1958 Frankem Rosenblattem. Tyto modely byly však omezené a nebyly schopné řešit složité úlohy jako například rozpoznávání obrazu nebo překlad textu, jak už jsme dnes zvyklí. (Rosenblatt, 1957)

V posledních několika desetiletích však došlo k velkému vývoji v počítačovém průmyslu, kdy došlo k razantnímu nárůstu výpočetního výkonu a zároveň s tím byly vynalezeny nové metody strojového učení. Neuronové sítě se tak staly klíčovým nástrojem k řešení řady obtížných úloh. Tyto úlohy by se daly v počítačovém vidění rozdělit na 4 základní typy: klasifikace snímku, detekce objektů, sémantická segmentace a segmentace instancí. (Tian, Han, Wang, Guan, Gu & Wei, 2022) (Dargan, Kumar, Ayyagari, & Kumar, 2020) V této práci se zabýváme primárně úkolem segmentace, kterou si rozebereme níže.

V průběhu let se objevilo několik základních typů architektur: Auto-Encoder (AE), Konvoluční neuronové sítě (CNN), Restricted Boltzmann Machine (RBM), Deep Stacking Network (DSN), Long Short Term Memory (LSTM)/Gated Recurrent Unit (GRU) network, Rekurentní neuronové sítě (RNN). Z těchto zmíněných základních architektur jsou nejběžněji používány CNN a LSTM. (Tian et al., 2022)

3.1. Klasifikace obrazu

Klasifikace obrazu je jedním ze základních úkolů, které řešíme pomocí neuronových sítí. Jde o úlohu zařazení obrázku do kategorie, někdy říkáme také třídy. Třída je přiřazena celému snímku. Snímek může být zároveň označen více třídami. Při učení sítě určené ke klasifikaci se síť snaží zobecnit rysy snímků, které jsou jí předkládány a na základě těchto rysů predikuje pravděpodobnosti jednotlivých kategorií, které umí model rozpoznat. (Dhillon & Verma, 2020) (Alzubaidi et al., 2021)

3.2. Detekce objektů v obraze

Při detekci objektů v obraze se oproti klasifikaci obrazu do kategorie kvůli časové náročnosti využívá metody klouzavého okna, což je základní technika pro detekci objektů. Tato metoda využívá okno o smysluplné velikosti, toto okno je posouváno po obraze a jednotlivý obsah je klasifikátorem ohodnocen kladně nebo negativně, podle toho, zda se v okně nachází objekt zájmu či ne. Techniky, které využívají této metody, lze rozdělit do dvou kategorií - „region proposal-based“, která označuje oblasti, kde se objekty nachází, mezi ty se řadí R-CNN nebo Faster R-CNN (Zhao et al., 2019), a „classification-based“, kam spadá síť YOLO (Redmon, Divvala, Girshick, & Farhadi) nebo SSD (Liu et al., 2016). (Dhillon & Verma, 2020)

3.3. Konvoluční neuronové sítě

Konvoluční neuronové sítě jsou hlavním typem architektury, kterou se tato práce zabývá. Je inspirována způsobem zpracování zrakového vjemu v mozkové kůře zvířat. Obvykle se taková neuronová síť skládá z mnoha vrstev, kde v těch prvních vrstvách dochází k rozpoznávání hlavních rysů, jako jsou třeba hrany. Ve vrstvách blízkých výstupu sítě se tyto rysy kombinují do abstraktních atributů následovaných samotnou klasifikací. V konvolučních neuronových sítích nevystupují pouze klasické konvoluční vrstvy, ale jsou kombinovány pooling vrstvami, které snižují dimenzionalitu vstupu a přispívají výrazně k abstrakci dat. (Dargan, Kumar, Ayyagari, & Kumar, 2020)

3.3.1. Vrstvy v konvolučních neuronových sítích

Vstupní vrstva – Vstupní vrstva neboli „input layer“ je místem, kde do sítě vstupuje obraz, základní stavební jednotkou obrazu v digitálním světě je pixel. Hodnoty, kterých pixely mohou nabývat jsou celá čísla z intervalu od 0 včetně do 255 včetně. Jednotlivé pixely jsou seskládány do tvaru matice, těchto matic může být i více, potom mluvíme o více kanálech obrazu.

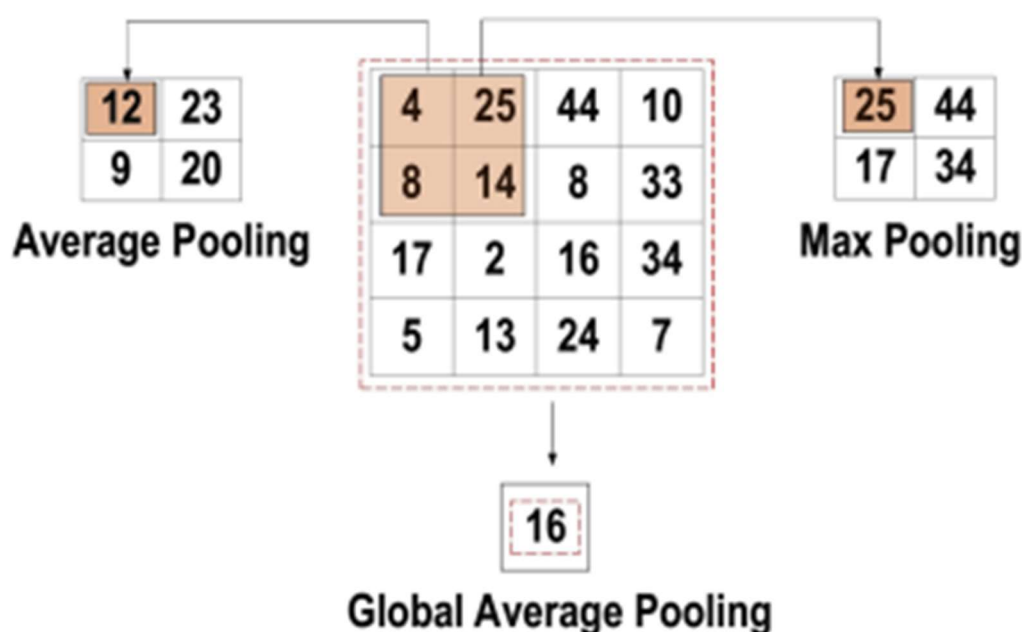
Nejčastěji se hovoří o 3 kanálovém obraze ve formátu RGB. Každá jedna hodnota pixelu je pak přivedena na vstup nějakého neuronu v síti, stejně jak je tomu u propojení lidského oka s centry v mozku. (Bhatt et al., 2021)

Konvoluční vrstva – Konvoluční vrstva je velmi důležitou součástí architektury CNN. Na její vstup je přivedena matice hodnot. Vrstva aplikuje filtry typu 3x3, 5x5 nebo jiné

a provede operaci konvoluce. Operace konvoluce je naznačena na obrázku. Její důležitou schopností je automaticky extrahovat charakteristické rysy z obrázku. (Bhatt et al., 2021) (Alzubaidi et al., 2021)

Pooling vrstva – Hlavním úkolem poolovacích vrstev v CNN je snižování dimenze výstupních tenzorů z konvolučních vrstev. Jedná se většinou o výběr jedné hodnoty z určitého výběru. Asi nejčastěji používaným je „Max Pooling“, ale existují i jiné metody výběru, ty nejčastější jsou ukázány na obrázku. (Alzubaidi et al., 2021) (Khanday & Sofi, 2021)

Obrázek 3: Typy pooling vrstev



Zdroj: Alzubaidi et al., 2021

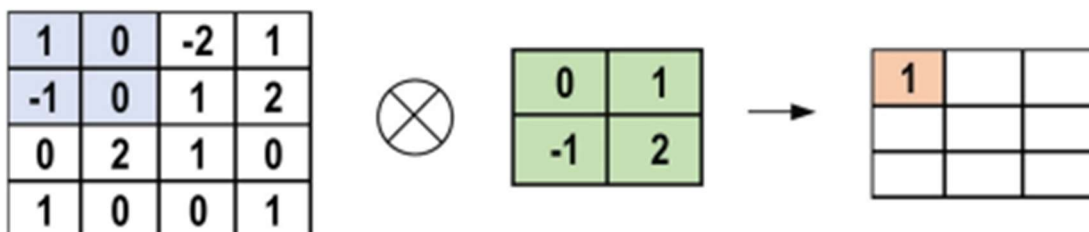
Plně propojená vrstva – Plně propojená nebo někdy lineární vrstva reprezentuje nějakou lineární transformaci dat. Říká se jí plně propojená, jelikož každý vstup této vrstvy je přiveden na každý neuron této vrstvy a nějakým způsobem ovlivňuje výstup daného neuronu. U moderních neuronových sítí se většinou tyto vrstvy nacházejí na konci sítě a většina parametrů je obsažena právě v těchto plně propojených vrstvách. (Bhatt et al., 2021) (Khanday & Sofi, 2021)

Nelineární vrstva (Aktivační funkce) – aktivační funkce je důležitá z hlediska mapování výstupních hodnot z předchozích vrstev do určitých intervalů, kterým potom dáváme nějaký význam, například -1 až 1, 0 až 1, a tak dále. V moderních neuronových

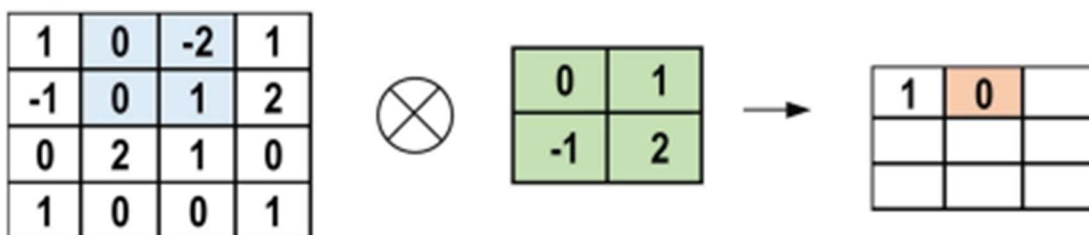
sítích se často používá oblíbená aktivační funkce ReLU nebo její různé variace. (Bhatt et al., 2021) (Alzubaidi et al., 2021) (Khanday & Sofi, 2021)

Obrázek 4: Operace konvoluce

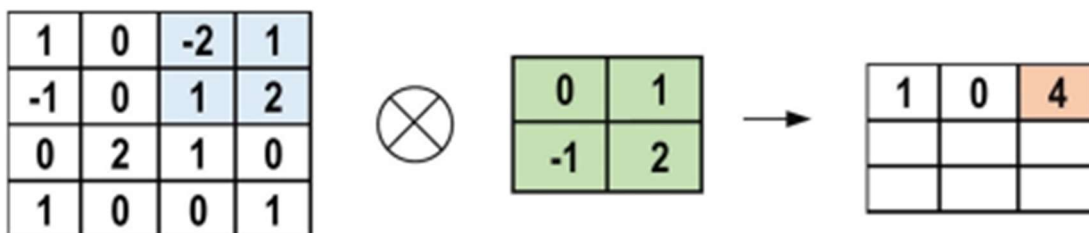
Step-1



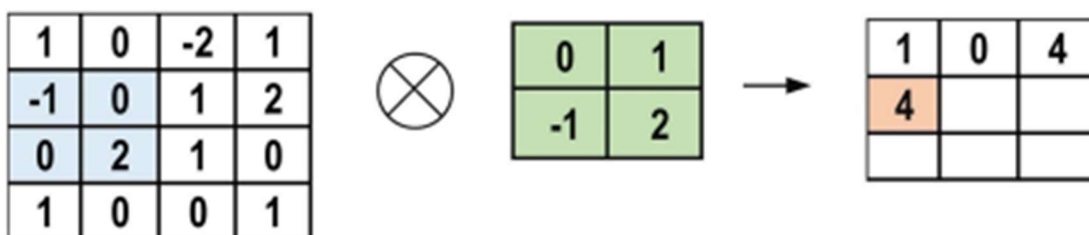
Step-2



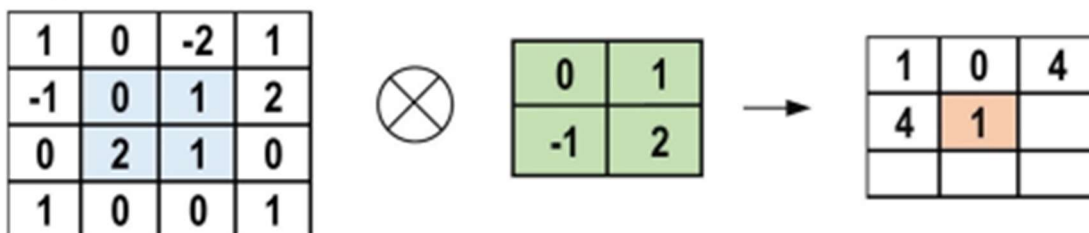
Step-3



Step-4



Step-5



Zdroj: Alzubaidi et al., 2021

3.3.2. Ztrátové funkce

Ztrátové funkce mají největší využití při učení neuronových sítí. Většina z nich jsou koncipovány jako dvou parametrové funkce, které přijímají jako parametr aktuální výstup neuronové sítě a požadovaný výstup neuronové sítě. Výstupem takové funkce je pak nějaká velikost chyby. Proto se pro každý typ úkolu hodí jiná ztrátová funkce. Velmi vhodnou ztrátovou funkcí pro klasifikaci obrazu do více tříd je například funkce Cross-Entropy nebo Softmax loss. Pro regresní problémy se hodí Euklidovská ztrátová funkce, jindy nazývaná také Mean square error (MSE). Pro sémantickou segmentaci se hodí také například Cross-Entropy loss, protože jde vlastně o klasifikaci jednotlivých pixelů do několika tříd. (Dhillon & Verma, 2020)

3.3.3. Metody optimalizace parametrů neuronové sítě

Neuronovou síť si můžeme představit jako soustavu n -dimenzionálních funkcí nebo transformací dat. Tato soustava dostane na vstup nějaká data a požadujeme po ní určitý výstup. Při prvotní inicializaci jsou váhy zvoleny náhodně, tedy výstup sítě není ani zdaleka blízky požadovanému výstupu. Porovnáním aktuálního výstupu s požadovaným výstupem získáme pomocí nějaké ztrátové funkce chybovou hodnotu, kterou se snažíme minimalizovat. Této ztrátové funkci se také někdy říká kritérium nebo cena. Učení neuronové sítě je tedy poměrně složitá optimalizační úloha, kterou značně usnadňují některé níže uvedené algoritmy a principy.

Algoritmus zpětné propagace – Při trénování neuronové sítě probíhají obecně dva kroky, *feed-forward* a zpětná propagace gradientu, což je postup, díky kterému zjistíme, jak se daná váha neuronu promítne s daným vstupem na výstup sítě. Této informace dále využívají optimalizační algoritmy, které hýbou následně s jednotlivými hodnotami vah podle zpětně propagovaného gradientu.

Parametry – Další hybnou silou při učení neuronové sítě jsou zejména dva parametry – *learning rate* a *batch size* – **Learning rate** je součinitel velikosti posunu parametrů podle gradientu. Jeho účelem je ovládání velikosti kroku při učení, aby nedocházelo k oscilacím kolem minima. Samozřejmě je zde stále riziko uvíznutí v nějakém lokálním minimu, to potom řeší některé optimalizační algoritmy. **Batch size** vyjadřuje kolik jednotlivých vzorků se zpracuje najednou, tím se docílí větší obecnosti při posunu v parametrech pomocí průměrování.

SGD – Stochastic gradient descent je algoritmus založený na náhodném výběru z dostupných gradientů, ze kterých se průměrováním vypočte výsledný gradient, podle kterého se potom posouvají váhy. V praxi se spíše používá upravená varianta s tzv. *Nesterov momentum*, která přináší do algoritmu určitou setrvačnost z předchozích iterací, která pomáhá překonat lokální minima. (Ruder, 2017)

RMSprop – *Root Mean Square propagation* je algoritmus, který využívá klouzavého průměru kvadrátů k eliminaci velkých rozdílů mezi velikostmi jednotlivých kroků pro danou váhu. (Ruder, 2017)

Adam – je asi v dnešní době nejčastěji používaným algoritmem. Je odvozen od SGD, ale využívá k urychlení klesání exponenciálně vážený průměr momentů z předchozích kroků. (Ruder, 2017) (Zhang, 2018)

3.3.4. Metody vyhodnocování výsledků

Při vyhodnocování výsledků jsou sledovány řady metrik, které se uzpůsobují dané úloze. Využívají se také při porovnávání výkonu různých architektur mezi sebou na obecně známých datasetech.

Mezi obecně aplikovatelné se řadí několik základních metrik. Přesnost (Accuracy) vyjadřuje poměr správně označených vzorků ku celkovému počtu vyhodnocovaných vzorků. (Alzubaidi et al., 2021) Citlivost (Recall nebo Sensitivity) je vyjadřována pro každou třídu zvlášť, jde o poměr mezi správně označenými vzorky patřící do dané skupiny ku celkovému počtu vzorků, které byly sítí označeny, že do skupiny patří. Vyjadřuje správnost pozitivních označení. (Alzubaidi et al., 2021) Účinnost nebo specifická (Specificity) je počítána pro každou třídu zvlášť a vyjadřuje správnost negativní označení obdobně jako citlivost pro pozitivní označení. (Alzubaidi et al., 2021) Přesnost (Precision) vyjadřuje správnost sítě pro danou třídu. Definována je jako poměr správně označených ku počtu všech, které k dané třídě náleží. (Alzubaidi et al., 2021) Při klasifikaci do více tříd se nejčastěji používá matice záměn, ve které jsou zaneseny správné detekce i záměny mezi třídami. Ideální matice záměn má nenulová čísla pouze na hlavní diagonále.

Některé úlohy jsou ovšem složitější a vyžadují komplexnější metriky. IoU (Intersection over Union) je metrika, která umožňuje porovnání masek nebo bounding boxů. Je vyjádřena jako poměr mezi průnikem a sjednocením dvou oblastí, skutečným

označením, které odpovídá realitě, a označením, které vygenerovala síť. (Wu, Sahoo, & Hoi, 2020) Metrikou pro detekce více objektů na jednom snímku je mAP (mean average precision), která kombinuje různé metriky mezi třídami a objekty pomocí průměru. (Wu, Sahoo, & Hoi, 2020) F1 score a J score jsou metriky, které vznikají kombinací základních metrik zmíněných výše. (Alzubaidi et al., 2021)

3.3.5. Známé modely konvolučních neuronových sítí

V níže uvedeném seznamu jsou shrnuty čteně užívané modely neuronových sítí.

- **AlexNet** – jedná se o konvoluční síť o 8 vrstvách, kde 5 jich je konvolučních a 3 jsou plně propojené. Takto sestavená síť má přibližně 60 milionů trénovatelných parametrů. Novinkou, kterou tato síť přináší je použití ReLU aktivační funkce místo dříve typických funkcí sigmoid a tanh. (Dhillon & Verma, 2020) (Bhatt et al., 2021) (Alzubaidi et al., 2021)
- **VGG-16** – o něco později byla představena další konvoluční síť s 13 konvolučními a 3 plně propojenými vrstvami s aktivační funkcí ReLU. Oproti AlexNet používá také menší konvoluční jádra 2x2 a 3x3. Díky tomu má 138 milionů parametrů. Existuje i složitější verze VGG-19. (Benali Amjoud & Amrouch, 2020) (Bhatt et al., 2021) (Alzubaidi et al., 2021)
- **GoogLeNet** – přístup této sítě je značně jiný, zavádí tzv. „block of inception“, tento blok provádí konvoluční transformace na různě přeškálovaných datech, obsahuje několik různě velkých filtrů 1x1, 3x3, 5x5. Uprostřed sítě se využívá konvolučního jádra 1x1, aby se snížila dimenzionalita. Namísto plně propojených vrstev se také využívá „average pooling“ vrstev. Celkově má tato síť 22 vrstev a 5 milionů parametrů. Je základem pro síť YOLOv1. (Benali Amjoud & Amrouch, 2020) (Alzubaidi et al., 2021)
- **ResNet** – S postupem času a přidávání vrstev se zjistilo, že přesnost takovýchto sítí bude brzy na svém maximu a jakmile se ho dosáhne, přesnost začne opět klesat. Při zjištění tohoto faktu přišli autoři sítě ResNet s novými technikami, jak se této saturaci vyhnout. Těmito technikami jsou tzv. „residual connections“ a „skip connections“, které propojují výstup z předchozí vrstvy do vrstvy následující, kde dojde ke složení výstupu z těchto dvou předchozích vrstev do vstupu vrstvy následující. Dojde tak k přenosu detailnější informace

do hlubších vrtech. Zároveň se používají normalizační techniky. (Benali Amjoud & Amrouch, 2020) (Dhillon & Verma, 2020) (Alzubaidi et al., 2021)

- **Inception-ResNet-V2** – tato síť využívá „*block of inception*“ a zároveň zavádí techniky představené v síti ResNet. Při kratším učení je síť efektivnější než ResNet, ale při dlouhém učení zase zaostává. (Bhatt et al., 2021) (Alzubaidi et al., 2021)
- **DarkNet-19** – tato síť byla vyvíjena s myšlenkou, aby byla menší a zároveň stejně efektivní. Je založena na všech dříve zmíněných technikách. Skládá se z 19 konvolučních a 5 poolingových vrstev. Je použita například i v síti YOLOv2. (Benali Amjoud & Amrouch, 2020) (Dhillon & Verma, 2020) (Alzubaidi et al., 2021)
- **SqueezeNet** – tato architektura je doporučována pro low-power zařízení, která většinou nedisponují velkým množstvím paměti. Tato síť zabere pouhých 4.9 MB a za provozu vyžaduje jen 100 MB RAM. (Dargan, Kumar, Ayyagari, & Kumar, 2020) (Bhatt et al., 2021) (Alzubaidi et al., 2021)

3.4. Příklady aplikací ve vybraných oborech

V dnešním světě je kladen důraz hlavně na rychlost, spolehlivost, cenu a automatizaci. Proto se v nemalé řadě případů dnes sahá po automatizaci za pomoci počítačových nebo robotických systémů. Tyto systémy se snaží zrychlovat a zefektivnit procesy nebo se vyvarovat chybám. Níže jsou uvedeny příklady takových aplikací.

3.4.1. Aplikace v dopravě

Na denním pořádku se setkáváme s dopravními kamerami. Ty jsou většinou rozmístěny podle jejich účelu. Některé hlídají spouštění semaforů, jiné hlídají rychlost jedoucích aut nebo auta počítají. Ve větších městech jsou dokonce tyto kamery společně se semaforami, závorami nebo dalšími senzory spojovány do komplexních systémů, které potom řídí dopravu (Baumgart & Burger, 2022) (Khanjary, 2013).

Čím dál více se v dnešní době objevují různé koncepty autopilotů, a samořídících se aut, většina těchto systémů je založena na zpracování obrazu neuronovými sítěmi. Příkladem může být studie SDC-Net (Abdou & Kamal, 2022), která řeší zpracování obrazu z více kamer najednou pomocí IoT platformy a následné plánování trasy. Při autonomním řízení

je důležité dosahovat v reálném čase vysoké přesnosti, jednou z metod je re-kalibrování sémantiky (Mi et al., 2023).

Do kategorie dopravy spadá také například odhadování zpoždění spojů na základě dopravní situace. Zajímavou oblastí jsou predikce zpoždění u vlaků (Li, Huang, Wen, Jiang, & Rodrigues, 2022), kde je řešena nejen dopravní situace na železnici (interakce mezi jednotlivými vlaky), ale i velikosti stanic, počasí nebo historie zpoždění.

3.4.2. Průmyslové aplikace

V průmyslu jsou neuronové sítě využívány ke specifickým úkolům, speciálně ve výrobních procesech, jde o predikci možných budoucích chyb z měření různých veličin. Dále se může jednat o automatické inspekce součástek (Cao, Wang, Qi, Wang, & Jiang, 2019), detekce defektů při výrobě (Djavadifar, Graham-Knight, Körber, Lasserre, & Najjaran, 2022) (Sampath et al., 2023). V letectví nebo v energetice se pak využívá k detekcím trhlin nebo slabých míst v materiálech, které jsou používány na lopatky turbín, křídla nebo jiné součásti (Sreeshan, Dinesh, & Renji, 2020) (Waqar Akram et al., 2019).

3.4.3. Aplikace v lékařství

Z lékařských aplikací se často objevuje nejen zpracování různých ultrazvukových snímků, ale i další typy úkolů jako je identifikace pacientů s různými nemocemi, jednou z nich může být třeba revmatoidní artritida (Wasilewska, Pauk, & Ihnatouski, 2018) nebo rakovina prsu (Silva et al., 2016) nebo plic, dále lze třeba v kombinaci s termokamerami monitorovat nakažené chřipkou (Lahiri, Bagavathiappan, Jayakumar, & Philip, 2012) nebo v posledních letech s onemocněním covid-19.

3.4.4. Zemědělské aplikace

Zemědělství aktuálně prochází velkými změnami a umělá inteligence včetně neuronových sítí je toho součástí. Ať už jde o automatické skleníky s detekcí plevelů (Maksimovich, Fedorov, & Kalichkin, 2023) (Fathipoor, Shah-Hosseini, & Arefi, 2023) nebo o hlídání zdraví zvířat (Xiong et al., 2019) nebo o analýzu jejich chování a přizpůsobení podmínek ve stájích. Neuronové sítě jsou využívány i k automatizaci zavlažování (Li et al., 2019) nebo ke kontrole kvality potravin (Izquierdo et al., 2020). V lesích nebo na lukách se uplatňuje detekce zvěře. (Dhillon & Verma, 2020)

4. Segmentace obrazu

Segmentace obrazu je oblast digitálního zpracování obrazu, která souhrnně označuje skupinu metod, které jsou založeny na různých principech, které mají za cíl automatické rozčlenění obrazu na oblasti, které mají nějaký smysluplný význam. Ve většině případů jde o identifikaci popředí a nalezení oblastí, které odpovídají významnému prvku zachycenému na snímku. (Dougherty, 2009)

4.1. Konvenční metody segmentace obrazu

4.1.1. Treshold-based algoritmy

Treshold-based algoritmy neboli algoritmy založené na prahových hodnotách jsou principiálně založeny na klasifikaci jednotlivých pixelů pomocí prahování jejich hodnoty v odstínech šedé. Pixely s hodnotou ve stejném intervalu jsou klasifikovány jako jedna kategorie, a považují se za oblast se stejnými vlastnostmi. Metoda je dobrá pro černobílé obrázky, které mají rovnoměrné rozložení v rámci odstínů šedé, a kde je dobře znatelná hranice mezi pozadím a objekty. Metoda je také náchylná na šum. (Tian et al., 2022)

4.1.2. Edge-based algoritmy

Jedná se o algoritmy, které využívají detekce pixelů na rozhraní v obraze. Podstatou těchto algoritmů je předpoklad, že každý jeden objekt na obrázku je ohraničený čarami, které jsou rozlišitelné podle intenzity obrazu. (Tian et al., 2022)

Při provádění hranově orientované segmentace se obvykle aplikuje hranový filtr, který pro každý pixel určuje, zda pixel náleží nějaké hraně nebo ne. Filtry jsou většinou založené na derivacích obrazu. Nejznámějším hranovým detektorem je Cannyho hranový detektor (Canny, 1986).

4.1.3. Cluster-based algoritmy

Algoritmy založené na klastrování využívají postupného spojování sousedních pixelů na základě podobnosti jejich vlastností. Nakonec jsou rozděleny do několika kategorií, tím pak vznikají hranice mezi oblastmi. (Tian et al., 2022) Mezi běžné metody řadíme K-means (Ayeche & Ziou, 2015), FCM (Noordam, van den Broek, & Buydens, 2000) nebo SLIC (Achanta, Shaji, Smith, Lucchi, Fua & Süsstrunk, 2012).

4.2. Neuronové sítě pro segmentaci

S rozvojem počítačového vidění a s mnohem vyššími nároky, které klade dnešní svět, s ohledem na inteligentní řízení, monitorování bezpečnosti se stávají tradiční metody nedostatečnými. Když vezmeme v potaz to, že deep learning se dostává do mnoha oblastí, tak stejný pokrok udělala i oblast segmentace obrazu. Algoritmy jako FCN nebo Mask R-CNN založené na hlubokém strojovém učení dostaly segmentační technologie do popředí v detekci objektů. Tyto technologie jsou schopné realizovat velmi přesnou detekci objektů v běžných scénériích, stále zde ale zůstává problém, že reálný svět je velmi proměnlivý a ovlivňuje ho mnoho faktorů, které neumíme ovlivnit. (Tian et al., 2022)

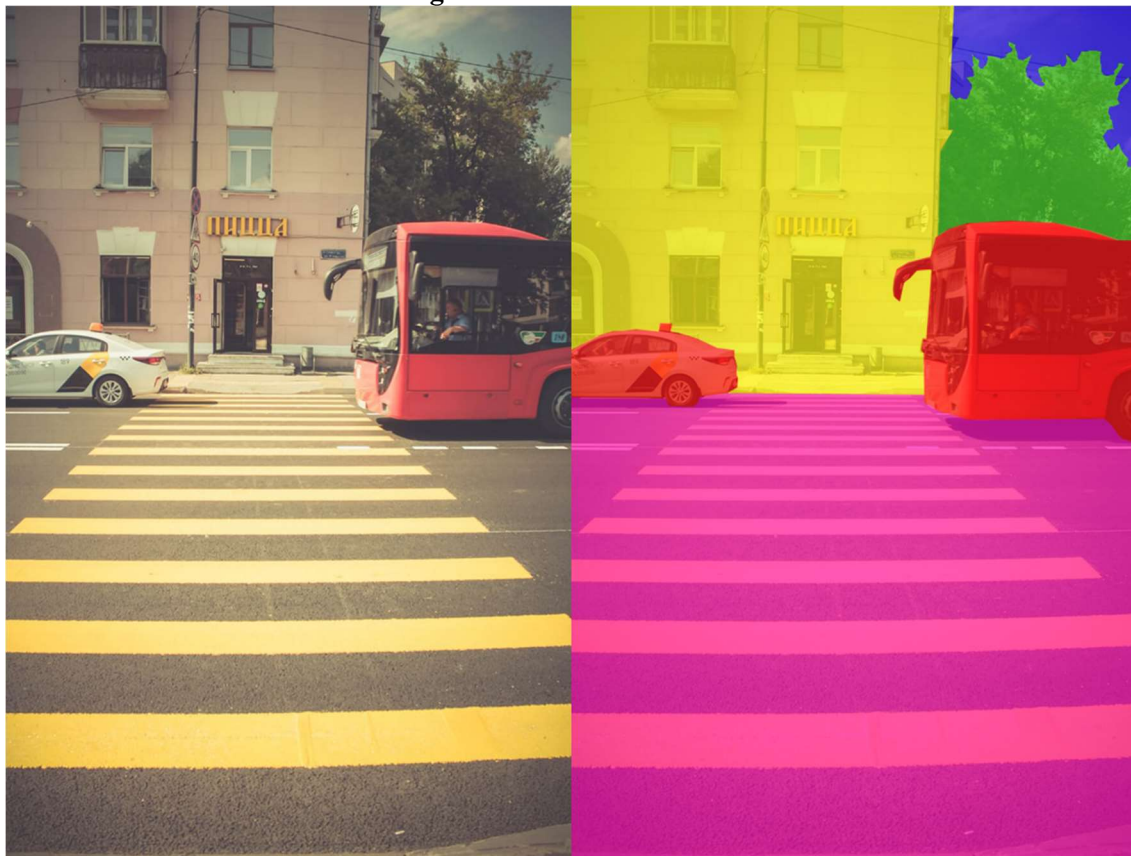
Z aktuálních výzkumů v oblasti segmentace pomocí neuronových sítí lze vyčíst, že jsou uznávány tři hlavní přístupy. První způsob přistupuje k segmentaci podobně jako k detekci objektů, a to tak, že ve dvou fázích postupně detekuje oblasti pro každý objekt v obraze a potom pro každý objekt tuto oblast rozdělí na masku objektu a pozadí. Druhý přístup je také dvoufázový a v principu kategorizuje každý pixel zvlášť, a nakonec pomocí klastrování pixely seskupuje. V poslední době se ale hodně rozmáhá jedno fázový přístup, který je z hlediska rychlosti doručení výsledku lepší. (Tian et al., 2022)

4.3. Typy segmentačních úkolů

4.3.1. Sémantická segmentace

Sémantická segmentace je proces rozdělení snímku na oblasti, které spolu významově souvisí. Sémantická segmentace se zaměřuje na rozlišení typů objektů nebo oblastí na základě jejich vlastností, například: typu povrchu, osoby, dopravní prostředky a další. Cílem je sestavit mapu pokrytí pro každý typ objektu nebo oblasti. Mapa vyjadřuje, které části snímku patří k jakému typu oblasti nebo objektu. Sémantická segmentace se hojně využívá při automatické analýze obrazu nebo videa, například pro řízení dopravy, robotiku nebo bezpečnostní aplikace. (Alzubaidi et al., 2021) (Dong, Wang, & Abbas, 2021) (Li & Chen, 2022)

Obrázek 5: Ukázka sémantické segmentace

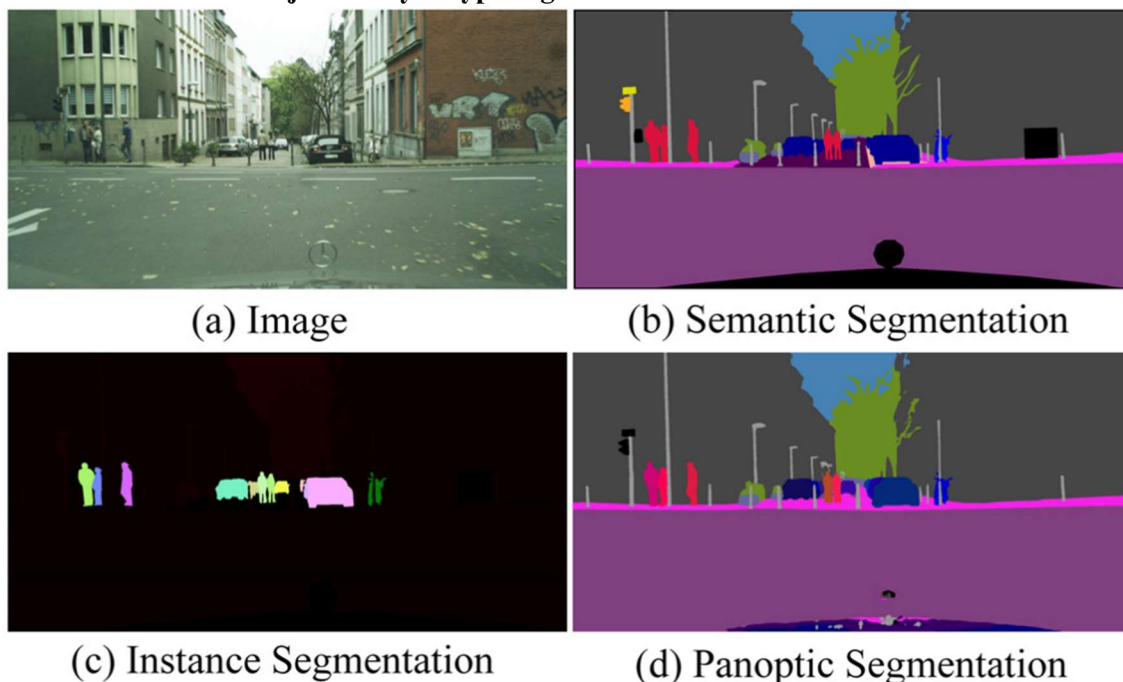


Zdroj: Anonymous, 2016

4.3.2. Segmentace instancí

Segmentace instancí je proces rozlišování a označování jednotlivých objektů nebo instancí na snímku. Segmentace instancí může být shrnuta jako doručení výsledku úkolů detekce objektů a sémantické segmentace v jednu chvíli naráz. Často se tak potýkáme se situacemi, že se jednotlivé objekty překrývají nebo že je v obraze velmi složité pozadí. (Tian et al., 2022) (Dong, Wang, & Abbas, 2021) V případě termografických snímků jde o rozlišování mezi jednotlivými tepelnými zdroji na snímku, například lidí nebo zvířat. Cílem je vytvořit mapu pokrytí pro každou instanci. Mapa pokrytí označuje, které části snímku patří k dané instanci. Tento proces je často používán jako součást předzpracování dat pro další analýzu. Segmentace instancí může poskytovat bohatší informaci oproti ostatním základním typům úkolů, které se v počítačovém vidění řeší. (Alzubaidi et al., 2021) (Dong, Wang, & Abbas, 2021) (Li & Chen, 2022)

Obrázek 6: Porovnání jednotlivých typů segmentace



Zdroj: Li & Chen, 2022

Segmentace instancí pomocí neuronových sítí bývá často spojována s tradičními metodami segmentace obrazu, které rozdělují obraz na nesouvislé ale smysluplné oblasti, ve kterých jednotlivé pixely určitým způsobem korelují. Mezi tradiční metody se řadí metody založené na prahových hodnotách, hranách a klastrování. (Tian et al., 2022) (Li & Chen, 2022)

4.3.3. Panoptická segmentace

Panoptická segmentace je kombinací segmentace instancí a sémantické segmentace, ale liší se v přístupu k jednotlivým typům objektů. Nejlépe se to dá asi přiblížit tak, že jako pozadí jsou označené věci z reálného světa, které nepočítáme, jako například obloha, budovy, tráva, silnice, ... Objekty jako osoby, auta, a další jsou brány jako popředí a je mezi nimi rozlišováno. (Li & Chen, 2022)

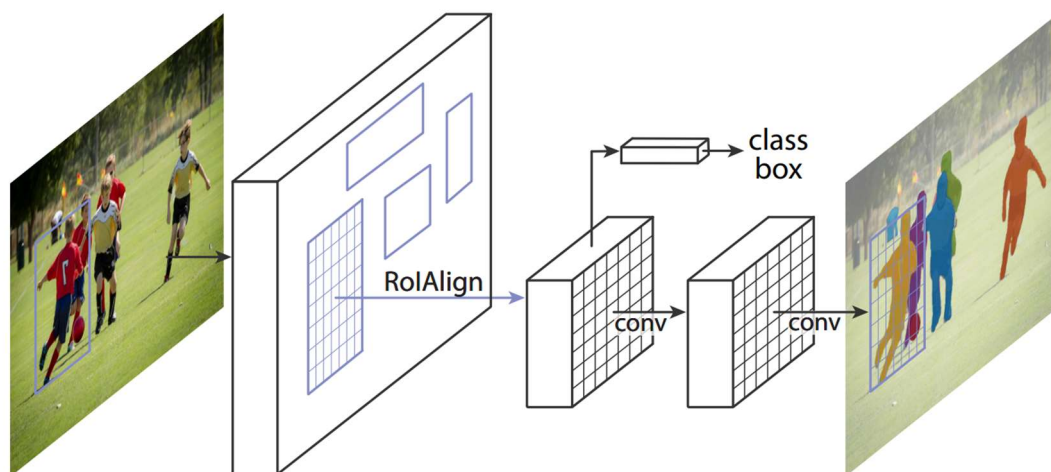
4.4. Segmentace instancí za pomoci neuronových sítí

4.4.1. Mask R-CNN

Model Mask R-CNN (He, Gkioxari, Dollar, & Girshick, 2020) byl publikován v roce 2018 a ve své době překonával všechny dosud známé modely v různých úkolech. Tento

model je relativně jednoduchý, flexibilní a lze ho snadno učit. Přesto model na svém výstupu poskytuje velmi kvalitní masky jednotlivých instancí. Dostupná je i verze s principem Faster RCNN – Mask R-CNN v tomto případě přidává paralelní větev k detekci masky. Mimo segmentaci instancí Mask R-CNN provádí klasickou detekci objektů nebo rozeznává pózy lidí.

Obrázek 7: Mask R-CNN



Zdroj: He, Gkioxari, Dollar, & Girshick, 2020

4.4.2. YOLACT

Model založený na detekčním modelu YOLO nazvaný YOLACT (Bolya, Zhou, Xiao, & Lee, 2019) má plně konvoluční topologii. Jedná se o rychlou a jednoduchou síť, v době, kdy byla představena, šlo o nejrychlejší síť tohoto druhu a používala se převážně k segmentaci v reálném čase. Rychlost byla dosažena díky paralelizaci dvou pod úkolů: generování prototypů masek a predikce koeficientů prototypů masky pro každou instanci. Masky jsou poté generovány jako lineární kombinace koeficientů a prototypů masek.

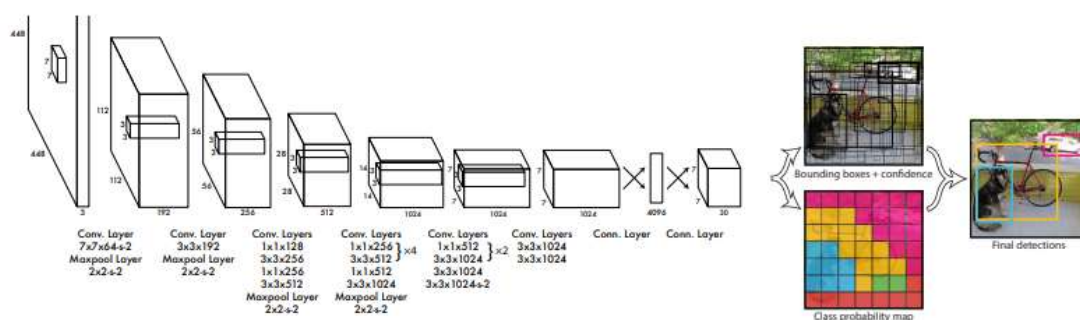
4.4.3. TensorMask

Síť označovaná jako TensorMask (Chen, Girshick, He, & Dollar, 2019) je založená na principu klouzavého okna s častými a pravidelnými mezerami. V souvislosti se segmentací instancí je tato oblast poměrně málo prozkoumaná. Pro každé umístění je spočítán trojrozměrný tenzor, pro reprezentaci predikce masky je poté využíván čtyřdimenzionální tenzor. Z hlediska přesnosti a výkonnosti vykazuje podobné výsledky jako Mask R-CNN.

4.4.4. YOLOv7

YOLOv7 (Wang, Bochkovskiy, & Liao, 2022) je nejaktuálnější verzí modelu YOLO, je vybudován na základě YOLOR, která měla za cíl přinést jednotnou a obecnou síť pro všechny typy úkolů. YOLOR vychází z YOLOv5, která je implementována ve frameworku PyTorch pro praktické využití. YOLOv7 prošlo oproti svému předchůdci reparametrizací a menšími strukturálními změnami. Ty měly za výsledek snížení počtu parametrů a malé zlepšení ve výkonnosti.

Obrázek 8: Základní architektura YOLO

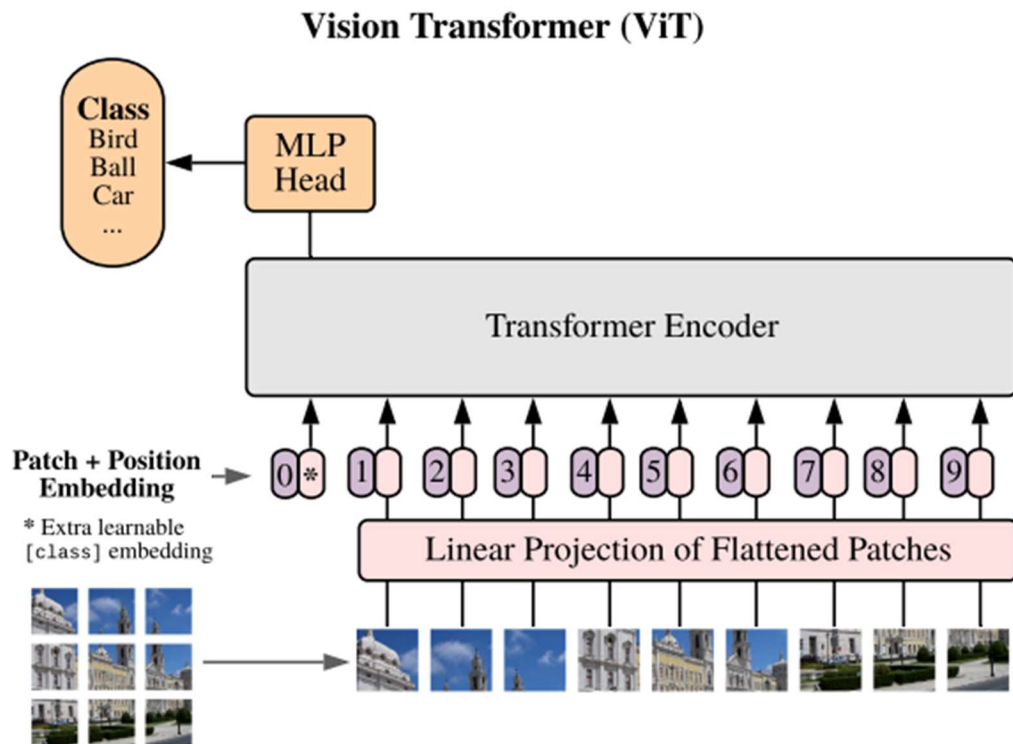


Zdroj: Zivkovic, 2021

4.4.5. EVA

EVA (Fang et al., 2022) je model představený v roce 2022, který ve všech směrech vyčnívá z řady, má nespočet parametrů, ve výkonnosti i velikosti obsazuje první příčky ve všech uznávaných benchmarcích. Model byl učený na veřejně dostupných datech. Architektura sítě EVA je škálovaný základní ViT (Vision Transformer). Ten funguje tak, že vstupní obrázek je rozdělen menší části, ty jsou pak paralelně transformátorem enkódovány. Při škálování se autoři dostali až k 1 miliardě parametrů, tomu odpovídají i provozní požadavky.

Obrázek 9: Vision Transformer

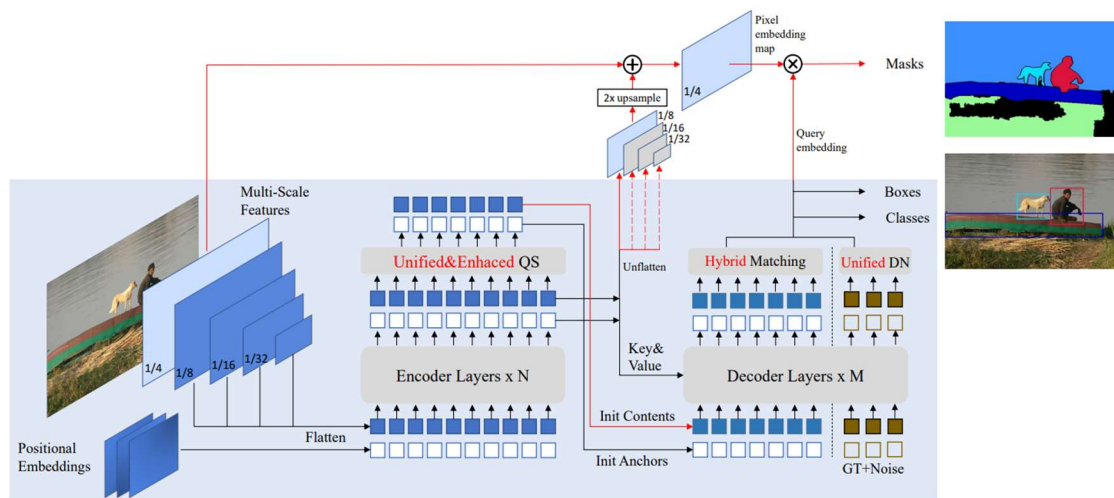


Zdroj: Dosovitskiy et al., 2021

4.4.6. Mask DINO

Mask DINO (Li et al., 2022) je vytvořena z detekční sítě DINO, je jejím rozšířením. Původní síť DINO je složena ze dvou hlavních větví, jedna zajišťuje predikci příslušnosti ke třídě a druhá predikci bounding boxů. Mask DINO do architektury přidává další větev, která zajišťuje predikci masek. Přidává i další klíčové komponenty, které jsou důležité pro segmentační úlohy.

Obrázek 10: Architektura sítě Mask DINO



Zdroj: Li et al., 2022

5. Příprava datasetu

5.1.1. Computer Vision Annotation Tool

CVAT (OpenCV, 2023) (Computer Vision Annotation Tool) je volně dostupný, open source nástroj, vyvíjený počítačovou společností Intel. Tento nástroj slouží vývojářům a datovým analytikům k přípravě rozsáhlých datasetů pro výzkum a vývoj praktických aplikací strojového vidění. Nástroj je dostupný v online webové verzi, tak i k sestavení a provozování na vlastním počítači. (OpenCV, 2023)

Tento nástroj jsme v práci také využili, jelikož poskytuje širokou škálu poloautomatických nástrojů pro tvorbu anotací, od klasických nástrojů využívajících prahování nebo jiných jednodušších algoritmů až po předpřipravené modely neuronových sítí, které pomohou s anotací a ručně už se jen udělají korekce. Dále třeba umožňuje paralelní práci více pracovníků na stejném datasetu nebo vyexportovat dataset do různých formátů.

5.1.2. Formát datasetu

Formátem datasetu se rozumí definovaná struktura uložení vzorků a jejich anotací, která se může velmi lišit mezi jednotlivými implementacemi. Každý framework má většinou definovaný svůj vlastní formát nebo využívá nějaký, který je dnes chápán už jako standard.

Formát datasetu byl volen podle potřeb jednotlivých trénovaných sítí. Základním formátem, který jsme využili byl formát COCO, který byl popsán společností Microsoft v době vzniku prvního datasetu od Microsoftu MSCOCO – Microsoft Common Objects in Context, kdy se později stal rozšířeným a vyvíjeným standardem. Z tohoto formátu bylo potřeba pro některé sítě dataset přetransformovat do jiného formátu, abychom nemuseli dělat velké zásahy do zdrojového kódu projektů vybraných sítí.

5.1.3. Obsah datasetu

Dataset, který jsme vytvořili v rámci této práce, obsahuje různé termografické snímky z prostředí stáje. Jedná se o několik typů snímků, např.: snímky z větší vzdálenosti, kde je vidět několik končetin zároveň, ale primárně se jedná o snímky jednotlivých paznehtů z různých úhlů. Celkově dataset obsahuje 1227 snímků s 2479 označenými objekty,

26 snímků je bez jediného objektu. Tento dataset byl následně rozdělen na tři části: trénovací (843 snímků s 1776 objekty), testovací (242 snímků s 463 objekty) a validační (121 snímků s 240 objekty), jako je běžné v poměru 7:2:1.

5.1.4. Tvorba anotací datasetu

CVAT nabízí různé typy anotací, pro segmentaci instancí je ideální polygonální maska, která je definována buďto relativně nebo absolutně k rozměrům snímku. Jde o sekvenci jednotlivých dvoudimenzionálních bodů, které definují vrcholy daného polygonu. V anotační aplikaci se vrcholy polygonu určují kliknutím na požadované místo ve snímku, přičemž se polygon průběžně vykresluje.

Touto technikou jsme postupně vytvořili všechny anotace pro všech 1227 snímků a po dokončení jsme dataset vyexportovali do zvoleného formátu COCO. Na následujících obrázcích 11, 12 a 13 jsou vidět typické snímky, které obsahuje námi vytvořený dataset. Anotace jednotlivých paznehtů jsou na obrázcích znázorněny tmavým překrytím.

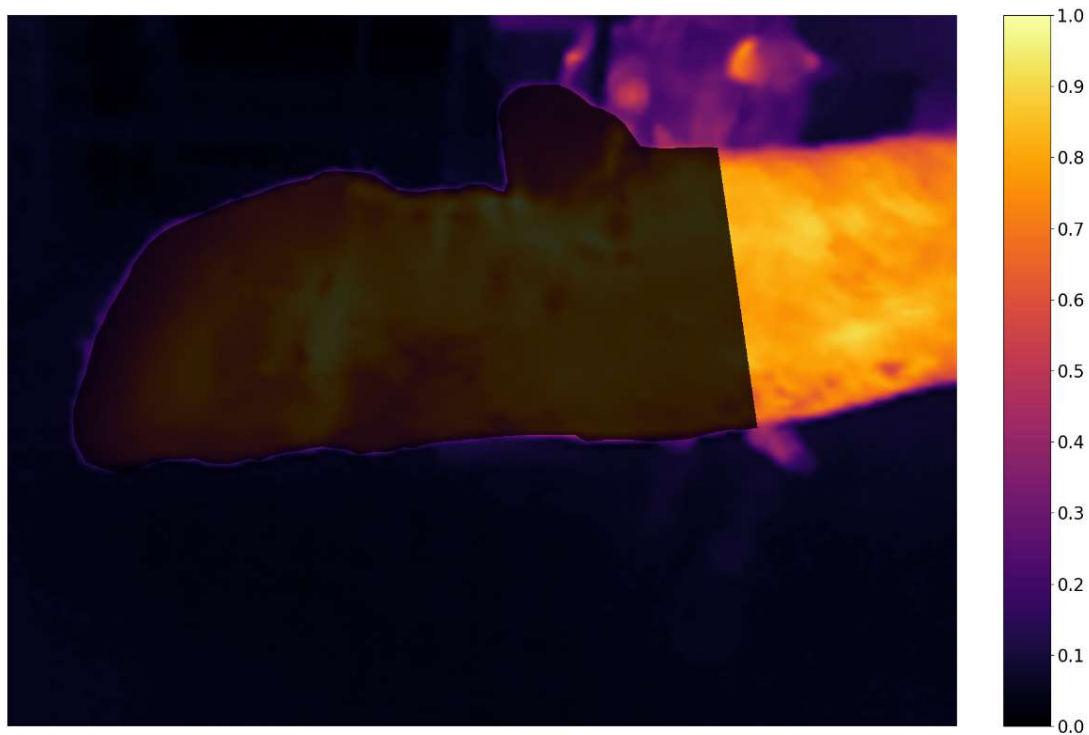
Obrázek 11: Ukázka datasetu – snímek z větší dálky



Obrázek 12: Ukázka datasetu – snímek paznehtu zředu



Obrázek 13: Ukázka datasetu – snímek paznehtu z boku



6. Učení neuronových sítí

Pro naši aplikaci jsme vybrali několik sítí z různých rodin, které vyhovovaly našim požadavkům a zároveň jejich náročnost odpovídala naší testovací sadě, která se skládá z procesoru AMD Ryzen 7 5700X pracující na frekvenci 3.4 GHz a grafické karty NVIDIA GeForce RTX 3060 s 3584 CUDA jádry a 12 GB VRAM doplněné o 32 GB RAM. Z výše zmíněných sítí byly vybrány Mask RCNN, YOLOv7 a Mask DINO. Tyto sítě budeme porovnávat pomocí metrik zmíněných v sekci 3.3.4, zejména pomocí mAP.

Velmi zajímavou síť EVA jsme nebyly schopni otestovat kvůli její vysoké paměťové náročnosti.

6.1.1. Mask R-CNN

Síť Mask R-CNN byla vybrána, protože se jedná o jednu z prvních úspěšných sítí v oblasti segmentace instancí v obraze a je považována za určitý standard, který je implementován v některých frameworkcích jako je *PyTorch* nebo *detectron2*.

Při přípravě této sítě jsme využili obou frameworků, přičemž první z nich jsme využili pro otestování a druhý z nich jsme využili k plnohodnotnému experimentu. Pomocí frameworku *detectron2* a předefinovaných modelů jsme připravili konfiguraci Mask R-CNN a nechali model při trénování projít 25000 iterací. Na následujících obrázcích 14 a 15 je vidět konfigurace modelu a hlavní smyčka s registrací našeho vlastního datasetu, který je užit při trénování.

Obrázek 14: Ukázka hlavního programu

```
from config import custom_config
from data import load_data
from detectron2.data import MetadataCatalog
from detectron2.data.datasets import register_coco_instances
from detectron2.engine import DefaultTrainer

if __name__ == '__main__':
    for d in ["train", "valid", "test"]:
        register_coco_instances(d, {}, "../../dataset/Hoofs/"+d+".json",
                               "../../dataset/Hoofs/")
        MetadataCatalog.get(d).set(thing_classes=["Hoof"])
        metadata = MetadataCatalog.get("train")

        cfg = custom_config("../runs/train/custom")

        trainer = DefaultTrainer(cfg)
        trainer.resume_or_load(resume=False)
        trainer.train()
```

Obrázek 15: Detectron2 vlastní konfigurace Mask R-CNN

```
from detectron2.config import get_cfg
from detectron2 import model_zoo

def custom_config(output_dir):
    cfg = get_cfg()

    # get configuration from model_zoo
    cfg.merge_from_file(model_zoo.get_config_file("COCO-
InstanceSegmentation/mask_rcnn_R_101_FPN_3x.yaml"))
    cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-
InstanceSegmentation/mask_rcnn_R_101_FPN_3x.yaml")

    # Model
    cfg.MODEL.MASK_ON = True
    cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1

    # Solver
    cfg.SOLVER.BASE_LR = 0.0002
    cfg.SOLVER.MAX_ITER = 40000
    cfg.SOLVER.STEPS = (20, 10000, 20000)
    cfg.SOLVER.gamma = 0.5
    cfg.SOLVER.IMS_PER_BATCH = 4

    # Test
    cfg.TEST.DETECTIONS_PER_IMAGE = 10

    # INPUT
    cfg.INPUT.MIN_SIZE_TRAIN = (800,)

    # DATASETS
    cfg.DATASETS.TEST = ('val',)
    cfg.DATASETS.TRAIN = ('train',)

    # DATASETS
    cfg.OUTPUT_DIR = output_dir

    return cfg
```

6.1.2. YOLOv7

Síť YOLOv7 byla vybrána z hlediska dlouhodobého prvenství v rychlosti a zároveň i v přesnosti ve všech oblastech zpracování obrazu. YOLOv7 je nejnovější verzí sítě rodiny YOLO, ve které se od verze YOLOv5 objevují větve k segmentaci instancí.

Při učení této sítě jsme postupovali podle instrukcí v oficiálním projektu ve veřejném verzovacím systému, kde autoři sítě uveřejnili včetně skriptů k přetrénování na vlastním datasetu. Síť YOLOv7 v uveřejněné podobě nepodporuje COCO formát datasetu, museli jsme tedy dataset transformovat pomocí skriptu na obrázku 16 a doplnit o soubor s informacemi o datasetu a následně jsme mohli síť přetrénovat pomocí příkazu na obrázku 17.

Obrázek 16: Ukázka transformačního skriptu pro dataset

```
if __name__ == '__main__':
    output_path = './yolov7/seg/data/test/'
    ds = Dataset('./dataset/Hoofs/', './dataset/Hoofs/test.json')
    for i in range(len(ds)):
        img, target = ds.__getitem__(i)
        out_img_path = os.path.join(output_path, 'images', target["name"])
        output_label_path = os.path.join(output_path, 'labels',
os.path.splitext(target["name"])[0] + '.txt')
        cv2.imwrite(out_img_path, img)
        with open(output_label_path, "w") as f:
            for seg in target["polygons"]:
                f.write('0 ')
                for n in seg:
                    f.write(str(n))
                    f.write(" ")
            f.write('\n')
```

Obrázek 17: Příkaz k přetrénování sítě YOLOv7 z oficiálního repozitáře

```
python segment/train.py --data yaml --batch 16 --weights '' --cfg yolov7-
seg.yaml --epochs 200 --name yolov7-seg
```

6.1.3.Mask DINO

Síť Mask DINO byla vybrána, protože je zajímavá svojí konstrukcí, strukturou a vykazuje slušné výsledky.

Síť je implementována pomocí frameworku *detectron2* a je dostupná podobně jako YOLOv7 ve veřejném verzovacím systému. Repozitář s projektem Mask DINO v uveřejněné verzi nepodporuje vlastní datasety. Museli jsme tedy do hlavní smyčky programu doplnit několik příkazů k zaregistrování vlastního datasetu tak jak je to naznačeno na obrázku 18. Dále bylo potřeba podobně jako u Mask R-CNN upravit

konfiguraci modelu, která je vidět na obrázku 19 a nakonec spustit přetrénování modelu příkazem na obrázku 20.

Obrázek 18: Ukázka registrace vlastního datasetu do frameworku detectron2

```
from detectron2.data.datasets import register_coco_instances
for d in ["train", "valid", "test"]:
    register_coco_instances(d, {}, "../..../dataset/Hoofs/"+d+".json",
                           "../..../dataset/Hoofs/")
    MetadataCatalog.get(d).set(thing_classes=["Hoof"])
metadata = MetadataCatalog.get("train")
```

Obrázek 19: Konfigurace modelu Mask DINO

```
_BASE_: base.yaml
MODEL:
  META_ARCHITECTURE: "MaskDINO"
  SEM_SEG_HEAD:
    NAME: "MaskDINOHead"
    IGNORE_VALUE: 255
    NUM_CLASSES: 80
    LOSS_WEIGHT: 1.0
    CONVS_DIM: 256
    MASK_DIM: 256
    NORM: "GN"
    # pixel decoder
    PIXEL_DECODER_NAME: "MaskDINOEncoder"
    DIM_FEEDFORWARD: 1024
    NUM_FEATURE_LEVELS: 3
    TOTAL_NUM_FEATURE_LEVELS: 3
    IN_FEATURES: ["res2", "res3", "res4", "res5"]
    DEFORMABLE_TRANSFORMER_ENCODER_IN_FEATURES: ["res3", "res4", "res5"]
    COMMON_STRIDE: 4
    TRANSFORMER_ENC_LAYERS: 6
  MaskDINO:
    TRANSFORMER_DECODER_NAME: "MaskDINODecoder"
    DEEP_SUPERVISION: True
    NO_OBJECT_WEIGHT: 0.1
    CLASS_WEIGHT: 4.0
    MASK_WEIGHT: 5.0
    DICE_WEIGHT: 5.0
    BOX_WEIGHT: 5.0
    GIOU_WEIGHT: 2.0
    HIDDEN_DIM: 256
    NUM_OBJECT_QUERIES: 300
    NHEADS: 8
    DROPOUT: 0.0
    DIM_FEEDFORWARD: 2048
    ENC_LAYERS: 0
    PRE_NORM: False
    ENFORCE_INPUT_PROJ: False
    SIZE_DIVISIBILITY: 32
    DEC_LAYERS: 9 # 9+1, 9 decoder layers, add one for the loss on
    learnable query
    TRAIN_NUM_POINTS: 12544
    OVERSAMPLE_RATIO: 3.0
    IMPORTANCE_SAMPLE_RATIO: 0.75
    INITIAL_PRED: True
    TWO_STAGE: True
    DN: "seg"
    DN_NUM: 100
    INITIALIZE_BOX_TYPE: "bitmask"
    TEST:
      SEMANTIC_ON: False
      INSTANCE_ON: True
      PANOPTIC_ON: False
      OVERLAP_THRESHOLD: 0.8
      OBJECT_MASK_THRESHOLD: 0.25
  SOLVER:
    AMP:
      ENABLED: True
  TEST:
    EVAL_PERIOD: 5000
# EVAL_FLAG: 1
```

Obrázek 20: Příkaz k přetrénování Mask DINO

```
python train_net.py --num-gpus 1 --config-file  
./configs/hoof/custom_dino.yml SOLVER.IMS_PER_BATCH 1 SOLVER.BASE_LR 0.001
```

7. Výsledky trénování vybraných sítí

K hodnocení výkonu sítí byla využita testovací část datasetu, která nebyla sítí předložena při učení.

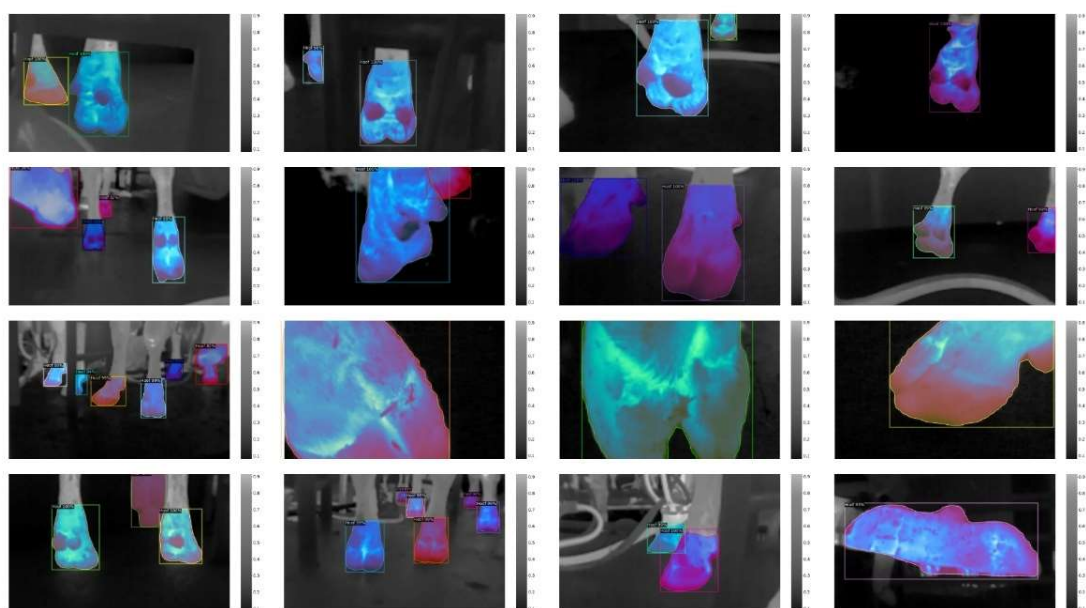
7.1. Vizualizace výsledků sítí

Na následujících obrázcích 21 a 22 jsou znázorněny predikce jednotlivých sítí na různých náhodně vybraných snímcích z testovací části datasetu, která nebyla předkládána sítím při učení. Vizualizace ze sítě Mask DINO uvedena není, jelikož není dostatečně přesvědčivá. Jelikož pouhým pohledem nejsme schopni exaktně vyhodnotit výkon sítě, přikládáme i tabulky se standardním vyhodnocením pomocí vyhodnocovacích nástrojů COCO v tabulkách 1, 2, 3, 4 a 5.

7.1.1. Mask R-CNN

U sítě Mask R-CNN jsme využili frameworku detectron2 a frameworku COCO, který poskytuje rozhraní pro využití standardních vyhodnocovacích metod a metrik. Pro vyhodnocení jsme využili IoU (Intersection over Union), už dříve zmíněnou metriku, a to jak pro přesnost detekce bounding boxů, tak i pro přesnost detekce masek jednotlivých objektů. Tyto metriky jsou v tabulkách 1 a 2, kde jsou v prvních 6 řádcích uvedeny přesnosti a v druhých 6 citlivosti, které jsou rozděleny podle atributů popsaných v tabulkách. Mezi tyto atributy se řadí procentuální překryv průniku se sjednocením predikce s anotací, velikost plochy, kterou predikce na snímku zabírá, ta je brána relativně k velikosti obrázku. Z tabulek se tedy dozvídáme, že se malé objekty v datasetu prakticky nenacházejí a že síť dobře registruje na paznehty, které jsou na obrázku zachyceny z blízka. Tabulky 1 a 2 také prozrazují, že u sítě Mask R-CNN, jsou detekce boxů a masek z hlediska výsledků prakticky stejné. Na vizualizaci si na obrázku 21 si můžeme všimnout, že u snímků z dálky se síti ne vždy podaří zaregistrovat pazneht celý, což se u snímků z blízka daří velmi dobře.

Obrázek 21: Vizualizace predikcí sítě Mask R-CNN



Tabulka 1: Výsledky sítě Mask R-CNN pro detekci bounding boxů

Average Precision @[IoU=0.50:0.95 area=all maxDets=15]	0.718
Average Precision @[IoU=0.50 area=all maxDets=15]	0.949
Average Precision @[IoU=0.75 area=all maxDets=15]	0.848
Average Precision @[IoU=0.50:0.95 area=small maxDets=15]	-1.00
Average Precision @[IoU=0.50:0.95 area=medium maxDets=15]	0.348
Average Precision @[IoU=0.50:0.95 area=large maxDets=15]	0.727
Average Recall @[IoU=0.50:0.95 area=all maxDets=1]	0.446
Average Recall @[IoU=0.50:0.95 area=all maxDets=10]	0.773
Average Recall @[IoU=0.50:0.95 area=all maxDets=15]	0.773
Average Recall @[IoU=0.50:0.95 area=small maxDets=15]	-1.00
Average Recall @[IoU=0.50:0.95 area=medium maxDets=15]	0.391
Average Recall @[IoU=0.50:0.95 area=large maxDets=15]	0.783

Tabulka 2: Výsledky sítě Mask R-CNN pro detekci masek instancí

Average Precision @[IoU=0.50:0.95 area=all maxDets=15]	0.738
Average Precision @[IoU=0.50 area=all maxDets=15]	0.949
Average Precision @[IoU=0.75 area=all maxDets=15]	0.894
Average Precision @[IoU=0.50:0.95 area=small maxDets=15]	-1.00
Average Precision @[IoU=0.50:0.95 area=medium maxDets=15]	0.248
Average Precision @[IoU=0.50:0.95 area=large maxDets=15]	0.748
Average Recall @[IoU=0.50:0.95 area=all maxDets=1]	0.453
Average Recall @[IoU=0.50:0.95 area=all maxDets=10]	0.786
Average Recall @[IoU=0.50:0.95 area=all maxDets=15]	0.786
Average Recall @[IoU=0.50:0.95 area=small maxDets=15]	-1.00
Average Recall @[IoU=0.50:0.95 area=medium maxDets=15]	0.427
Average Recall @[IoU=0.50:0.95 area=large maxDets=15]	0.795

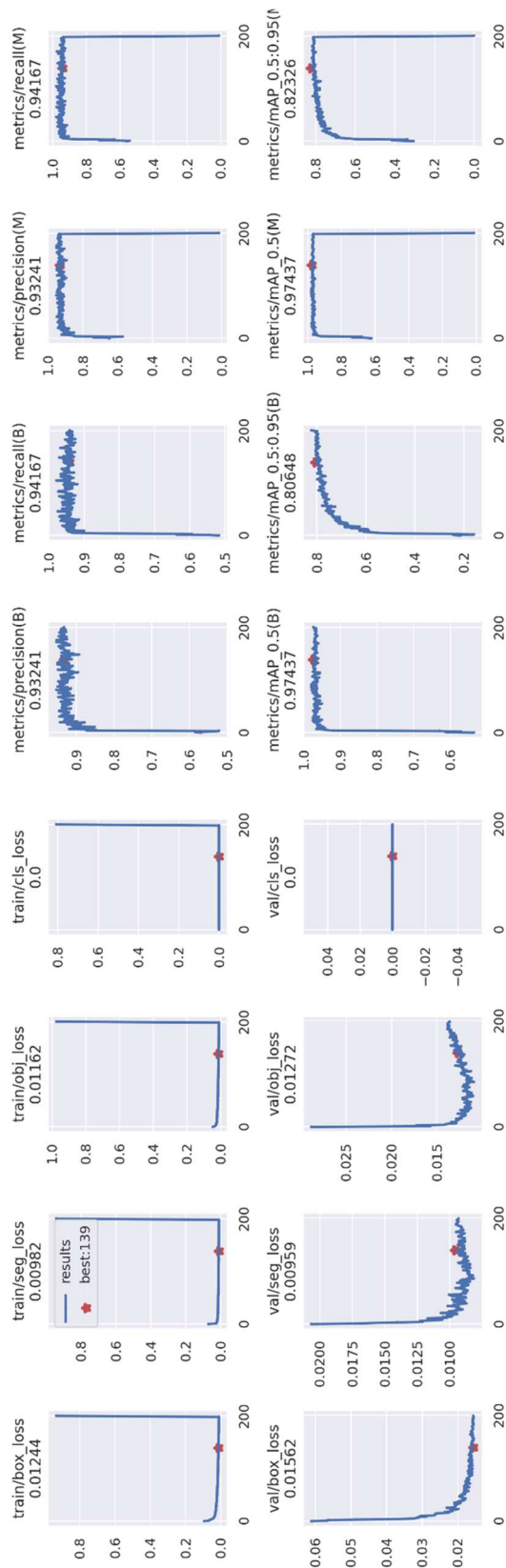
7.1.2. YOLOv7

V projektu sítě YOLOv7 byly dostupné pouze celkové vyhodnocení bez rozdělení na kategorie, v tabulce 3 jsou tedy uvedeny jen shrnující údaje, kde v prvních 4 řádcích tabulky jsou uvedeny metriky pro predikce bounding boxů objektů a v dalších 4 jsou uvedeny ty samé metriky pro příslušné masky. Metriky precision, recall i mAP jsou popsány v kapitole 3.3.4. Na obrázku 22 je v grafech uveden vývoj důležitých metrik pro trénování, především vývoj hodnot ztrátových funkcí. V těchto grafech je na vodorovné ose vyneseno číslo iterace a na ose y je uvedena hodnota dané metriky. Z těchto grafů můžeme vyčíst, že nejlepší výsledky podávala síť kolem 150 iterace, ze které jsou parametry modelu uloženy pro následné použití. Na obrázku 23 jsou znázorněny predikce vytvořené sítí YOLOv7 a je zde vidět, že síť je velmi přesná a zvládá i složitější kompozice, čemuž odpovídají i výsledky v tabulce 3.

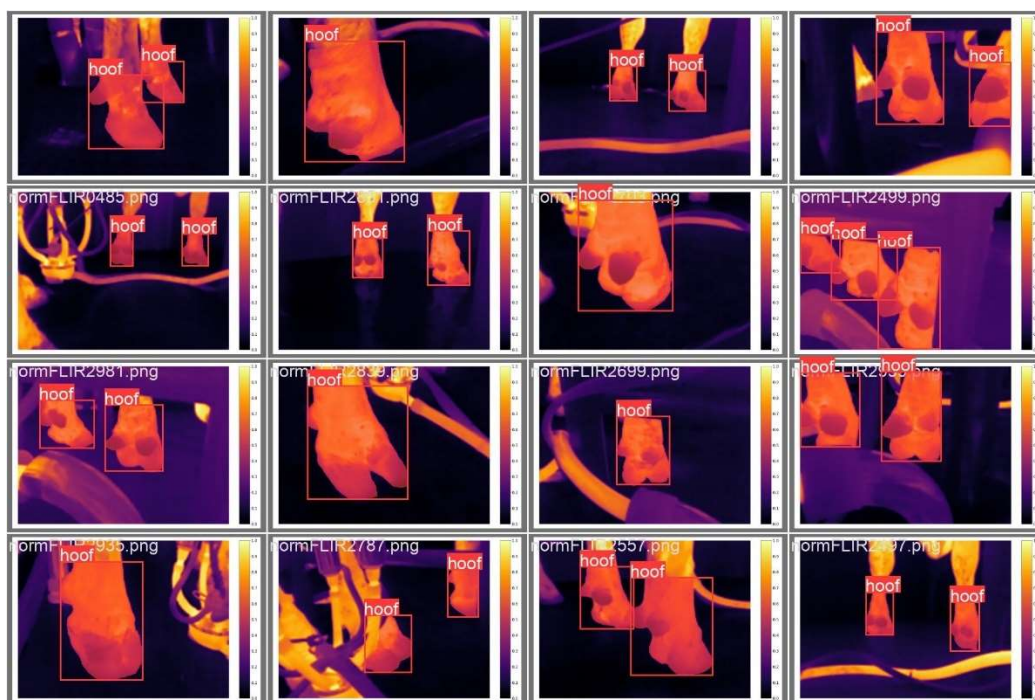
Na grafech na obrázcích 24 a 25 jsou vyobrazeny křivky udávající přesnost predikovaných boxů, respektive masek, v závislosti na jistotě, která udává, na kolik si je

síť jista, že jde o daný objekt. V obou grafech můžeme vidět, že přesnost se drží na velmi vysoké úrovni i když pozorujeme nižší míru jistoty.

Obrázek 22: YOLOv7 průběh trénování



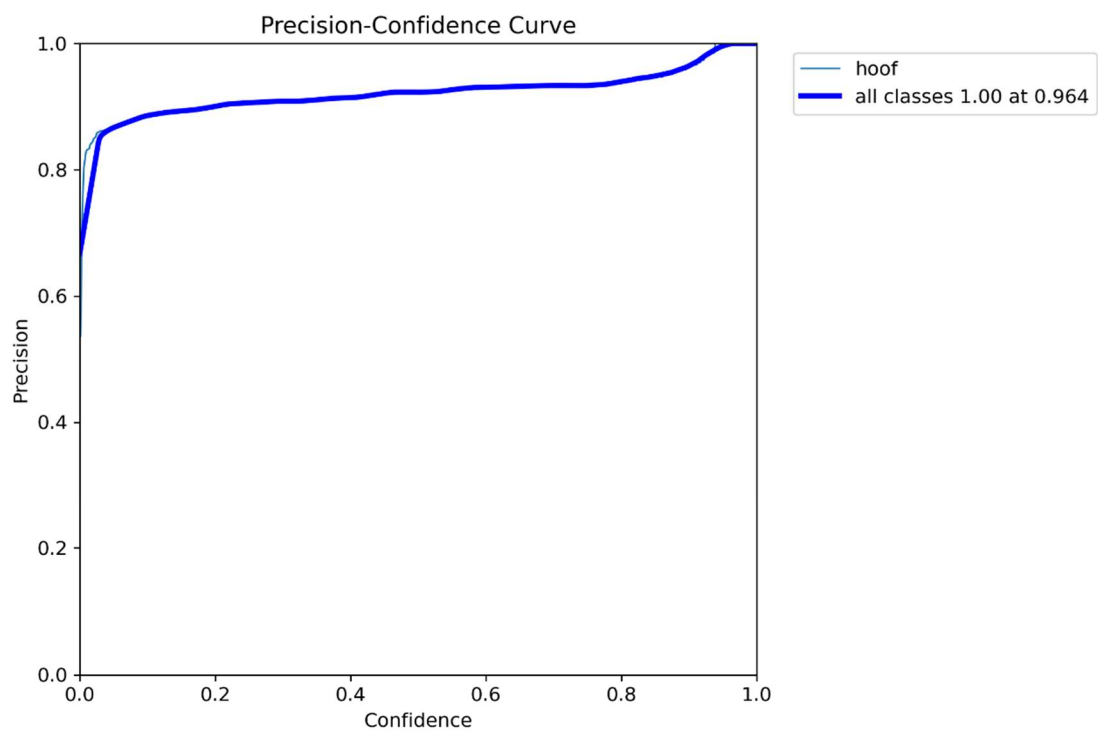
Obrázek 23: Vizualizace predikcí sítě YOLOv7



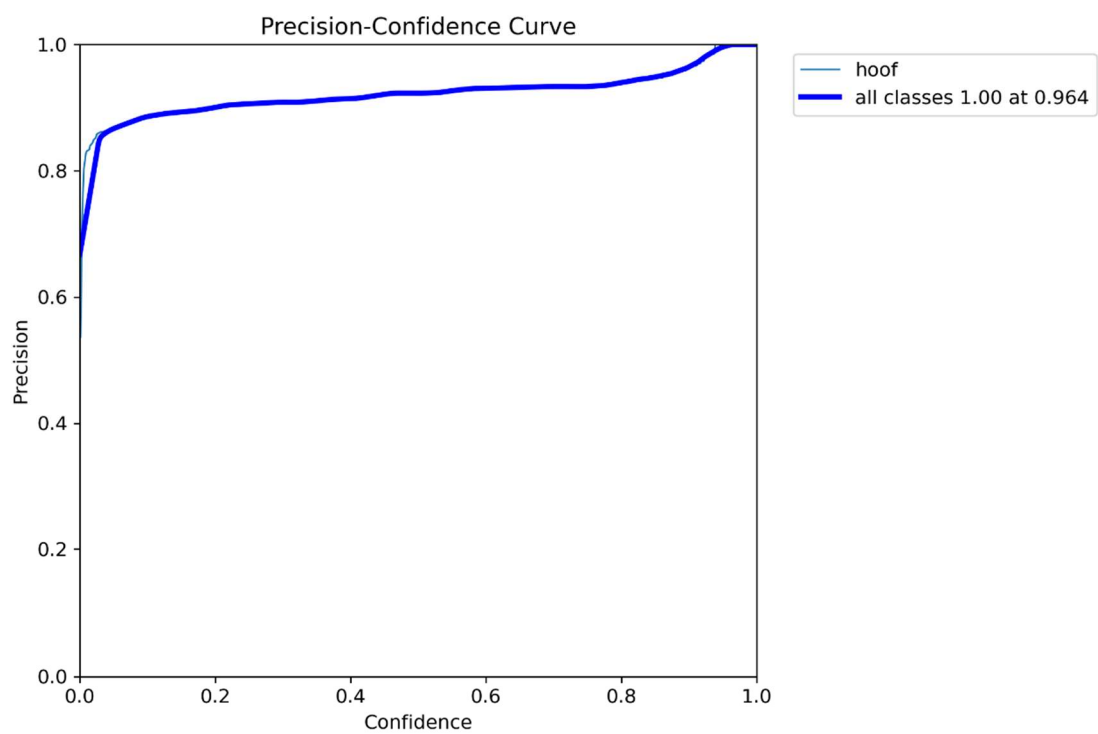
Tabulka 3: Vyhodnocení sítě YOLOv7

Box Precision	0.947
Box Recall	0.926
Box mAP50	0.962
Box mAP50-95	0.809
Mask Precision	0.947
Mask Recall	0.926
Mask mAP50	0.962
Mask mAP50-95	0.816
Inference time	3.7 ms

Obrázek 24: YOLOv7 Box P-C graf



Obrázek 25: YOLOv7 Mask P-C graf



7.1.3.Mask DINO

U sítě Mask DINO bylo použito stejného rozhraní jako u sítě Mask R-CNN. Tabulky jsou tedy stejného formátu a rozdělení, bohužel výsledky u této sítě nejsou takové, jaké byly očekávány při rešerši. U predikcí bounding boxů můžeme hovořit dokonce o absolutním selhání. U predikcí masek může jít i o náhodu. Z praktického hlediska, nám tabulky 4 a 5 prozrazují, že síť není schopna zaregistrovat v termosnímku pazneht vůbec.

Důvodů, které toto mohly způsobit může být mnoho. Například ten, že síť Mask DINO byla více paměťově náročná než zbylé dvě sítě a trénovala se na našem datasetu tedy pouze s velikostí sady 1. Zbylé dvě sítě byly trénovány s velikostí sady 10.

Tabulka 4: Výsledky sítě Mask DINO pro detekci bounding boxů

Average Precision @[IoU=0.50:0.95 area=all maxDets=15]	0.000
Average Precision @[IoU=0.50 area=all maxDets=15]	0.001
Average Precision @[IoU=0.75 area=all maxDets=15]	0.000
Average Precision @[IoU=0.50:0.95 area=small maxDets=15]	-1.00
Average Precision @[IoU=0.50:0.95 area=medium maxDets=15]	0.000
Average Precision @[IoU=0.50:0.95 area=large maxDets=15]	0.000
Average Recall @[IoU=0.50:0.95 area=all maxDets=1]	0.032
Average Recall @[IoU=0.50:0.95 area=all maxDets=10]	0.032
Average Recall @[IoU=0.50:0.95 area=all maxDets=15]	0.032
Average Recall @[IoU=0.50:0.95 area=small maxDets=15]	-1.00
Average Recall @[IoU=0.50:0.95 area=medium maxDets=15]	0.000
Average Recall @[IoU=0.50:0.95 area=large maxDets=15]	0.032

Tabulka 5: Výsledky sítě Mask DINO pro detekci masek instancí

Average Precision @[IoU=0.50:0.95 area=all maxDets=15]	0.004
Average Precision @[IoU=0.50 area=all maxDets=15]	0.013
Average Precision @[IoU=0.75 area=all maxDets=15]	0.001
Average Precision @[IoU=0.50:0.95 area=small maxDets=15]	-1.00
Average Precision @[IoU=0.50:0.95 area=medium maxDets=15]	0.000
Average Precision @[IoU=0.50:0.95 area=large maxDets=15]	0.004
Average Recall @[IoU=0.50:0.95 area=all maxDets=1]	0.195
Average Recall @[IoU=0.50:0.95 area=all maxDets=10]	0.195
Average Recall @[IoU=0.50:0.95 area=all maxDets=15]	0.195
Average Recall @[IoU=0.50:0.95 area=small maxDets=15]	-1.00
Average Recall @[IoU=0.50:0.95 area=medium maxDets=15]	0.000
Average Recall @[IoU=0.50:0.95 area=large maxDets=15]	0.200

7.2. Porovnání výsledků

V následující tabulce 6 jsou shrnuty výsledky a tučně jsou vyznačeny ty, které vykazují nejlepší výsledky v dané kategorii.

Tabulka 6: Porovnání výsledků

	Box mAP	Mask mAP
Mask R-CNN	0.718	0.715
YOLOv7	0.886	0.889
Mask DINO	0.010	0.361

V tabulce 6 je vidět, že původcem nejlepších predikcí na testovací části datasetu byla síť YOLOv7 a obstojně si vedla i síť Mask R-CNN. Síť Mask DINO se po experimentálním otestování stala prakticky nepoužitelnou.

Z hlediska paměťové a výkonnostní náročnosti je síť Mask DINO nejnáročnější. Na pomyslném druhém místě by se zařadila síť Mask R-CNN a vítězem by se stala také síť YOLOv7, která na naší experimentální sadě vykazovala velmi dobré výsledky i z hlediska rychlosti inferencí, kdy jeden obrázek dokázala zpracovat za 3,7 ms, což odpovídá přibližně 270 snímkům za vteřinu, u obou dalších byla rychlost znatelně nižší a pohybovala se v nižších desítkách snímků za vteřinu.

8. Závěr

V této práci jsme se zaměřili na termografické snímky a jejich zpracování pomocí konvolučních neuronových sítí, zejména nás zajímala úloha segmentace instancí, která je považována za jeden z náročnějších úkolů.

Cílem práce bylo připravit a otestovat několik modelů konvolučních neuronových sítí, které budou schopny rozeznat na termografickém snímku jednotlivé instance paznehtu skotu. Takto označené instance by mohly být dále využity k další analýze, která by ulehčila diagnostiku zdravotního stavu zvířete.

V první části práce vznikla obecná shrnující rešerše, ze které jsme zjistili, že využití termografických snímků je v medicíně a jiných oborech poměrně běžné. Dále jsme také zjistili, že jejich zpracování pomocí neuronových sítí se nijak zvlášť neliší od klasického zpracování obrazu. Nakonec byly vybrány 3 vhodné modely konvolučních neuronových sítí k natrénování. Mezi vybranými byly Mask R-CNN, YOLOv7 a Mask DINO. V rešerši jsme se seznámili také s několika zajímavými sítěmi, jako je například EVA, které mají ale značně vyšší nároky na hardware.

Ve druhé části práce vznikl kvalitní dataset termografických snímků paznehtů skotu, který obsahuje přes 1200 snímků a anotací ve formátu COCO. Dataset obsahuje snímky z různé vzdálenosti a z různých úhlů. Každý ze snímků má alespoň jednu anotaci. Snímky z větší dálky mají většinou anotací více.

V další části jsme vybrané modely učili na předpřipravených datech a vyhodnocovali jejich spolehlivost. Mezi třemi vybranými modely byly natrénovány síť Mask R-CNN, která dosahovala na necelých 72 % ve standardní metrice mAP pro hodnocení přesnosti masek, YOLOv7, která se v našem vyhodnocení pyšní mAP 89 %, a Mask DINO, která bohužel absolutně nesplnila předpoklady a dosáhla na pouhých 36 % mAP. Mohlo zde sehrát svoji roli také to, že Mask DINO při učení využívá velmi velké množství paměti, a proto se při učení využívalo nižší paralelizace zpracování.

Z hlediska rychlosti predikce se osvědčila velmi dobře síť YOLOv7, které stačilo k predikci na termografickém snímku 3,7 ms, což odpovídá přibližně 270 snímkům za vteřinu na naší testovací sadě. S výkonnějším hardwarem by mělo být možné dosáhnout i vyšší frekvence.

Z našeho výzkumu vyšla jako nejlepší volba pro segmentaci instancí paznehtu skotu síť YOLOv7. Špatnou volbou by nebyla ani Síť Mask R-CNN.

9. Summary and keywords

9.1. Summary

9.1.1. CZ

Tato práce se zabývá problémem segmentace instancí paznehtu skotu na termografických snímcích pomocí neuronových sítí. Cílem této práce bylo otestovat několik vybraných řešení a vyhodnocení jejich výsledků. Základem práce je rešerše existujících řešení, příprava datasetu, výběr modelů neuronových sítí a ohodnocení výsledků jednotlivých modelů. Práce popisuje postup prací a závěrem jsou porovnány jednotlivé výsledky a vyhodnoceno nejlepší řešení. Výsledkem práce je vytvořený dataset termografických snímků paznehtů a 3 otestované modely neuronových sítí, ze kterých 2 jsou velmi dobře použitelné pro řešení daného problému.

9.1.2. EN

This thesis deals with the issue of instance segmentation of cattle's hoof in thermographic images. The aim of this thesis was to test several solutions and evaluate them. The basis of this thesis is review of existing solutions and state-of-the-art in this field of study, dataset preparation, selection of neural network models and evaluation of the results of individual models. The thesis describes the progress of the work and in the conclusion the individual results are compared and the best solution is evaluated. As results of this thesis are the created dataset of thermographic images of cattle's hoofs and 3 tested and evaluated models of neural networks, from which 2 of them are useable as solution for this issue.

9.2. Keywords

9.2.1. CZ

Termografie, neuronové sítě, pazneht skotu, YOLOv7, Mask R-CNN, Mask DINO, segmentace instancí

9.2.2. EN

Thermography, neural nets, cattle's hoof, YOLOv7, Mask R-CNN, Mask DINO, instance segmentation

10. Seznam použitých zdrojů

- A. Al-Falluji, R., Dalaf Katheeth, Z., & Alathari, B. (2021). Automatic Detection of COVID-19 Using Chest X-Ray Images and Modified ResNet18-Based Convolution Neural Networks [Online], 66(2), 1301-1313. <https://doi.org/10.32604/cmc.2020.013232>
- Abdou, M., & Kamal, H. A. (2022). SDC-Net: End-to-End Multitask Self-Driving Car Camera Cocoon IoT-Based System [Online]. Sensors, 22(23). <https://doi.org/10.3390/s22239108>
- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC Superpixels Compared to State-of-the-Art Superpixel Methods [Online]. Ieee Transactions On Pattern Analysis And Machine Intelligence, 34(11), 2274-2282. <https://doi.org/10.1109/TPAMI.2012.120>
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., et al. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions [Online]. Journal Of Big Data, 8(1). <https://doi.org/10.1186/s40537-021-00444-8>
- Asdrubali, F., Baldinelli, G., Bianchi, F., Costarelli, D., Rotili, A., Seracini, M., & Vinti, G. (2018). Detection of thermal bridges from thermographic images by means of image processing approximation algorithms [Online]. Applied Mathematics And Computation, 317, 160-171. <https://doi.org/10.1016/j.amc.2017.08.058>
- Ayech, M. W., & Ziou, D. (2015). Automated feature weighting and random pixel sampling in k-means clustering for terahertz image segmentation [Online]. 2015 Ieee Conference On Computer Vision And Pattern Recognition Workshops (Cvprw), 35-40. <https://doi.org/10.1109/CVPRW.2015.7301294>
- Bagavathiappan, S., Lahiri, B. B., Saravanan, T., Philip, J., & Jayakumar, T. (2013). Infrared thermography for condition monitoring – A review [Online], 60, 35-55. <https://doi.org/10.1016/j.infrared.2013.03.006>
- Baumgart, U., & Burger, M. (2022). Optimal Control of Traffic Flow Based on Reinforcement Learning [Online]. Smart Cities, Green Technologies, And Intelligent Transport Systems, 313-329. https://doi.org/10.1007/978-3-031-17098-0_16

- Benali Amjoud, A., & Amrouch, M. (2020). Convolutional Neural Networks Backbones for Object Detection [Online]. *Image And Signal Processing*, 282-289. https://doi.org/10.1007/978-3-030-51935-3_30
- Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., et al. (2021). CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope [Online]. *Electronics*, 10(20). <https://doi.org/10.3390/electronics10202470>
- Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: Real-Time Instance Segmentation [Online]. 2019 *Ieee/Cvf International Conference On Computer Vision (Iccv)*, 9156-9165. <https://doi.org/10.1109/ICCV.2019.00925>
- Brodzicki, A., Jaworek-Korjakowska, J., Kleczek, P., Garland, M., & Bogyo, M. (2020). Pre-Trained Deep Convolutional Neural Network for *Clostridioides Difficile* Bacteria Cytotoxicity Classification Based on Fluorescence Images [Online]. *Sensors*, 20(23). <https://doi.org/10.3390/s20236713>
- Canny, J. (1986). A Computational Approach to Edge Detection [Online]. *Ieee Transactions On Pattern Analysis And Machine Intelligence*, PAMI-8(6), 679-698. <https://doi.org/10.1109/TPAMI.1986.4767851>
- Cao, J., Wang, P., Qi, Y., Wang, D., & Jiang, Y. (2019). Application research of infrared image processing technology in the quality inspection of riveting parts [Online]. *The Journal Of Engineering*, 2019(23), 9226-9229. <https://doi.org/10.1049/joe.2018.9221>
- Dargan, S., Kumar, M., Ayyagari, M. R., & Kumar, G. (2020). A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning [Online]. *Archives Of Computational Methods In Engineering*, 27(4), 1071-1092. <https://doi.org/10.1007/s11831-019-09344-w>
- Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies and applications to object detection [Online]. *Progress In Artificial Intelligence*, 9(2), 85-112. <https://doi.org/10.1007/s13748-019-00203-0>
- Djavadifar, A., Graham-Knight, J. B., Körber, M., Lasserre, P., & Najjaran, H. (2022). Automated visual detection of geometrical defects in composite manufacturing processes using deep convolutional neural networks [Online]. *Journal Of Intelligent Manufacturing*, 33(8), 2257-2275. <https://doi.org/10.1007/s10845-021-01776-1>

- Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications [Online]. *Computer Science Review*, 40. <https://doi.org/10.1016/j.cosrev.2021.100379>
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [Online]. <https://doi.org/10.48550/arXiv.2010.11929>
- Dougherty, G. (2009). Digital Image Processing for Medical Applications [Online]. <https://doi.org/10.1017/CBO9780511609657>
- Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., et al. (2022). EVA: Exploring the Limits of Masked Visual Representation Learning at Scale [Online]. <https://doi.org/10.48550/arXiv.2211.07636>
- Fathipoor, H., Shah-Hosseini, R., & Arefi, H. (2023). CROP AND WEED SEGMENTATION ON GROUND-BASED IMAGES USING DEEP CONVOLUTIONAL NEURAL NETWORK [Online]. *Isprs Annals Of The Photogrammetry, Remote Sensing And Spatial Information Sciences*, X-4/W1-2022, 195-200. <https://doi.org/10.5194/isprs-annals-X-4-W1-2022-195-2023>
- Fernández-Cuevas, I., Bouzas Marins, J. C., Arnáiz Lastras, J., Gómez Carmona, P. M., Piñonosa Cano, S., García-Concepción, M. Á., & Sillero-Quintana, M. (2015). Classification of factors influencing the use of infrared thermography in humans: A review [Online], 71, 28-55. <https://doi.org/10.1016/j.infrared.2015.02.007>
- Gong, W., Peng, Y., He, M., & Wang, J. (2015). Thermal image and spectral characterization of roadway failure process in geologically 45° inclined rocks [Online]. *Tunnelling And Underground Space Technology*, 49, 156-173. <https://doi.org/10.1016/j.tust.2015.04.011>
- He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2020). Mask R-CNN [Online]. *Ieee Transactions On Pattern Analysis And Machine Intelligence*, 42(2), 386-397. <https://doi.org/10.1109/TPAMI.2018.2844175>
- Chen, X., Girshick, R., He, K., & Dollar, P. (2019). TensorMask: A Foundation for Dense Object Segmentation [Online]. 2019 *Ieee/Cvf International Conference On Computer Vision (Iccv)*, 2061-2069. <https://doi.org/10.1109/ICCV.2019.00215>

- Izquierdo, M., Lastra-Mejías, M., González-Flores, E., Cancilla, J. C., Aroca-Santos, R., & Torrecilla, J. S. (2020). Deep thermal imaging to compute the adulteration state of extra virgin olive oil [Online]. *Computers And Electronics In Agriculture*, 171. <https://doi.org/10.1016/j.compag.2020.105290>
- Jones, B. F. (1998). A reappraisal of the use of infrared thermal image analysis in medicine [Online]. *Ieee Transactions On Medical Imaging*, 17(6), 1019-1027. <https://doi.org/10.1109/42.746635>
- Khanday, N. Y., & Sofi, S. A. (2021). Taxonomy, state-of-the-art, challenges and applications of visual understanding: A review [Online]. *Computer Science Review*, 40. <https://doi.org/10.1016/j.cosrev.2021.100374>
- Khanjary, M. (2013). Using game theory to optimize traffic light of an intersection [Online]. 2013 Ieee 14Th International Symposium On Computational Intelligence And Informatics (Cinti), 249-253. <https://doi.org/10.1109/CINTI.2013.6705201>
- Lahiri, B. B., Bagavathiappan, S., Jayakumar, T., & Philip, J. (2012). Medical applications of infrared thermography: A review [Online], 55(4), 221-235. <https://doi.org/10.1016/j.infrared.2012.03.007>
- Li, C., Du, R., Luo, Q., Wang, R., & Ding, X. (2023). A Novel Model of Thyroid Nodule Segmentation for Ultrasound Images [Online], 49(2), 489-496. <https://doi.org/10.1016/j.ultrasmedbio.2022.09.017>
- Li, F., Zhang, H., xu, H., Liu, S., Zhang, L., Ni, L. M., & Shum, H. -Y. (2022). Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation [Online]. <https://doi.org/10.48550/arXiv.2206.02777>
- Li, H., Yin, J., Zhang, M., Sigrimis, N., Gao, Y., & Zheng, W. (2019). Automatic diagnosis of strawberry water stress status based on machine vision [Online]. *International Journal Of Agricultural And Biological Engineering*, 12(1), 159-164. <https://doi.org/10.25165/j.ijabe.20191201.4293>
- Li, X., & Chen, D. (2022). A survey on deep learning-based panoptic segmentation [Online]. *Digital Signal Processing*, 120. <https://doi.org/10.1016/j.dsp.2021.103283>

- Li, Z., Huang, P., Wen, C., Jiang, X., & Rodrigues, F. (2022). Prediction of train arrival delays considering route conflicts at multi-line stations [Online]. *Transportation Research Part C: Emerging Technologies*, 138. <https://doi.org/10.1016/j.trc.2022.103606>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. -Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector [Online]. *Computer Vision – Eccv 2016*, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
- Long, F., Peng, J. -J., Song, W., Xia, X., & Sang, J. (2021). BloodCaps: A capsule network based model for the multiclassification of human peripheral blood cells [Online]. *Computer Methods And Programs In Biomedicine*, 202. <https://doi.org/10.1016/j.cmpb.2021.105972>
- Maksimovich, K., Fedorov, D., & Kalichkin, V. (2023). Feed Forward Neural Network Modelling for Spring Wheat Crop Forecast [Online]. *Agriculture Digitalization And Organic Production*, 303-312. https://doi.org/10.1007/978-981-19-7780-0_27
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity [Online]. *The Bulletin Of Mathematical Biophysics*, 5(4), 115-133. <https://doi.org/10.1007/BF02478259>
- Mi, A., Gao, M., Huo, Z., Qiao, Y., Chen, J., & Jia, H. (2023). Semantics recalibration and detail enhancement network for real-time semantic segmentation [Online]. *Iet Computer Vision*. <https://doi.org/10.1049/cvi2.12180>
- Min, D. -H., Jang, B. -su, & Yoon, H. -K. (2023). Application of infrared thermography for estimating residual stress in ground anchors for maintenance [Online]. *Scientific Reports*, 13(1). <https://doi.org/10.1038/s41598-022-27222-7>
- Noordam, J. C., van den Broek, W. H. A. M., & Buydens, L. M. C. (2000). Geometrically guided fuzzy C-means clustering for multivariate image segmentation [Online]. *Proceedings 15Th International Conference On Pattern Recognition. Icpr-2000*, 462-465. <https://doi.org/10.1109/ICPR.2000.905376>
- OpenCV. (2023) Computer Vision Annotation Tool (CVAT) [Online]. Retrieved from <https://github.com/opencv/cvat>

Osco, L. P., Marcato Junior, J., Marques Ramos, A. P., de Castro Jorge, L. A., Fatholahi, S. N., de Andrade Silva, J., et al. (2021). A review on deep learning in UAV remote sensing [Online]. *International Journal Of Applied Earth Observation And Geoinformation*, 102. <https://doi.org/10.1016/j.jag.2021.102456>

Pham, V., Pham, C., & Dang, T. (2020). Road Damage Detection and Classification with Detectron2 and Faster R-CNN [Online]. *2020 Ieee International Conference On Big Data (Big Data)*, 5592-5601. <https://doi.org/10.1109/BigData50022.2020.9378027>

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection [Online], 779-788. <https://doi.org/10.1109/CVPR.2016.91>

Rosenblatt, F. (1957). The perceptron - A perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory.

Ruder, S. (2017). An overview of gradient descent optimization algorithms [Online]. <https://doi.org/10.48550/arXiv.1609.04747>

Sampath, V., Murtua, I., Martin, J. J. A., Rivera, A., Molina, J., & Gutierrez, A. (2023). Attention Guided Multi-Task Learning for Surface defect identification [Online]. *Ieee Transactions On Industrial Informatics*, 1-9. <https://doi.org/10.1109/TII.2023.3234030>

Sanchez-Moreno, A. S., Olivares-Mercado, J., Hernandez-Suarez, A., Toscano-Medina, K., Sanchez-Perez, G., & Benitez-Garcia, G. (2021). Efficient Face Recognition System for Operating in Unconstrained Environments [Online]. *Journal Of Imaging*, 7(9). <https://doi.org/10.3390/jimaging7090161>

Santos, T. T., de Souza, L. L., dos Santos, A. A., & Avila, S. (2020). Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association [Online]. *Computers And Electronics In Agriculture*, 170. <https://doi.org/10.1016/j.compag.2020.105247>

Sarwar, F., Griffin, A., Rehman, S. U., & Pasang, T. (2021). Detecting sheep in UAV images [Online]. *Computers And Electronics In Agriculture*, 187. <https://doi.org/10.1016/j.compag.2021.106219>

Semantic Segmentation [Online]. Retrieved from https://upload.wikimedia.org/wikipedia/commons/d/d4/Image_segmentation.png

- Silva, L. F., Santos, A. A. S. M. D., Bravo, R. S., Silva, A. C., Muchaluat-Saade, D. C., & Conci, A. (2016). Hybrid analysis for indicating patients with breast cancer using temperature time series [Online]. *Computer Methods And Programs In Biomedicine*, 130, 142-153. <https://doi.org/10.1016/j.cmpb.2016.03.002>
- Sreeshan, K., Dinesh, R., & Renji, K. (2020). Nondestructive inspection of aerospace composite laminate using thermal image processing [Online]. *Sn Applied Sciences*, 2(11). <https://doi.org/10.1007/s42452-020-03619-9>
- Tian, D., Han, Y., Wang, B., Guan, T., Gu, H., & Wei, W. (2022). Review of object instance segmentation based on deep learning [Online]. *Journal Of Electronic Imaging*, 31(04). <https://doi.org/10.1117/1.JEI.31.4.041205>
- Wang, C. -Y., Bochkovskiy, A., & Liao, H. -Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors [Online]. <https://doi.org/10.48550/arXiv.2207.02696>
- Waqar Akram, M., Li, G., Jin, Y., Chen, X., Zhu, C., Zhao, X., et al. (2019). Improved outdoor thermography and processing of infrared images for defect detection in PV modules [Online]. *Solar Energy*, 190, 549-560. <https://doi.org/10.1016/j.solener.2019.08.061>
- Wasilewska, A., Pauk, J., & Ihnatouski, M. (2018). Image Processing Techniques for ROI Identification in Rheumatoid Arthritis Patients from Thermal Images [Online]. *Acta Mechanica Et Automatica*, 12(1), 49-53. <https://doi.org/10.2478/ama-2018-0008>
- Wen, H., Huang, C., & Guo, S. (2021). The Application of Convolutional Neural Networks (CNNs) to Recognize Defects in 3D-Printed Parts [Online]. *Materials*, 14(10). <https://doi.org/10.3390/ma14102575>
- Wu, Q., Feng, D., Cao, C., Zeng, X., Feng, Z., Wu, J., & Huang, Z. (2021). Improved Mask R-CNN for Aircraft Detection in Remote Sensing Images [Online]. *Sensors*, 21(8). <https://doi.org/10.3390/s21082618>
- Wu, X., Sahoo, D., & Hoi, S. C. H. (2020). Recent advances in deep learning for object detection [Online]. *Neurocomputing*, 396, 39-64. <https://doi.org/10.1016/j.neucom.2020.01.085>

- Xiong, X., Lu, M., Yang, W., Duan, G., Yuan, Q., Shen, M., et al. (2019). An Automatic Head Surface Temperature Extraction Method for Top-View Thermal Image with Individual Broiler [Online]. *Sensors*, 19(23). <https://doi.org/10.3390/s19235286>
- Yang, D., Wang, X., Zhang, H., Yin, Z. -yu, Su, D., & Xu, J. (2021). A Mask R-CNN based particle identification for quantitative shape evaluation of granular materials [Online]. *Powder Technology*, 392, 296-305. <https://doi.org/10.1016/j.powtec.2021.07.005>
- Zaridis, D. I., Mylona, E., Tachos, N., Pezoulas, V. C., Grigoriadis, G., Tsiknakis, N., et al. (2023). Region-adaptive magnetic resonance image enhancement for improving CNN-based segmentation of the prostate and prostatic zones [Online]. *Scientific Reports*, 13(1). <https://doi.org/10.1038/s41598-023-27671-8>
- Zhang, Y., Wang, Z., Song, R., Yan, C., & Qi, Y. (2022). Detection-by-tracking of traffic signs in videos [Online]. *Applied Intelligence*, 52(7), 8226-8242. <https://doi.org/10.1007/s10489-021-02838-w>
- Zhang, Z. (2018). Improved Adam Optimizer for Deep Neural Networks [Online]. 2018 Ieee/Acm 26Th International Symposium On Quality Of Service (Iwqos), 1-2. <https://doi.org/10.1109/IWQoS.2018.8624183>
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring [Online]. *Mechanical Systems And Signal Processing*, 115, 213-237. <https://doi.org/10.1016/j.ymssp.2018.05.050>
- Zivkovic, N. M. (2021). YOLO Approach [Online]. Retrieved from <https://rubikscode.net/wp-content/uploads/2021/04/yolo1.png>

11. Seznam obrázků a tabulek

Obrázek 1: Termografický snímek prsu	7
Obrázek 2: Termografický snímek domu	8
Obrázek 3: Typy pooling vrstev	11
Obrázek 4: Operace konvoluce	12
Obrázek 5: Ukázka sémantické segmentace	20
Obrázek 6: Porovnání jednotlivých typů segmentace	21
Obrázek 7: Mask R-CNN	22
Obrázek 8: Základní architektura YOLO	23
Obrázek 9: Vision Transformer	24
Obrázek 10: Architektura sítě Mask DINO	25
Obrázek 11: Ukázka datasetu – snímek z větší dálky	27
Obrázek 12: Ukázka datasetu – snímek paznehtu zpředu	28
Obrázek 13: Ukázka datasetu – snímek paznehtu z boku	28
Obrázek 14: Ukázka hlavního programu	30
Obrázek 15: Detectron2 vlastní konfigurace Mask R-CNN	30
Obrázek 16: Ukázka transformačního skriptu pro dataset	31
Obrázek 17: Příkaz k přetrénování sítě YOLOv7 z oficiálního repozitáře	31
Obrázek 18: Ukázka registrace vlastního datasetu do frameworku detectron2	32
Obrázek 19: Konfigurace modelu Mask DINO	33
Obrázek 20: Příkaz k přetrénování Mask DINO	34
Obrázek 21: Vizualizace predikcí sítě Mask R-CNN	36
Obrázek 22: YOLOv7 průběh trénování	38
Obrázek 23: Vizualizace predikcí sítě YOLOv7	39
Obrázek 24: YOLOv7 Box P-C graf	40
Obrázek 25: YOLOv7 Mask P-C graf	40

Tabulka 1: Výsledky sítě Mask R-CNN pro detekci bounding boxů	36
Tabulka 2: Výsledky sítě Mask R-CNN pro detekci masek instancí	37
Tabulka 3: Vyhodnocení sítě YOLOv7	39
Tabulka 4: Výsledky sítě Mask DINO pro detekci bounding boxů	41
Tabulka 5: Výsledky sítě Mask DINO pro detekci masek instancí	42
Tabulka 6: Porovnání výsledků	42

12. Seznam příloh

Příloha 1: Modely sítí (DVD)

Příloha 2: Zdrojový kód (DVD)

13. Přílohy