



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

APLIKACE PRO SPRÁVU FOTEK A EDITACI META-DAT

APPLICATION FOR MANAGING PHOTOS AND EDITING METADATA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PATRIK FRÜŠTÖK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. LUKÁŠ SEMERÁD, Ph.D.

BRNO 2024

Zadání bakalářské práce



161604

Ústav: Ústav inteligentních systémů (UITS)
Student: **Frůštök Patrik**
Program: Informační technologie
Název: **Aplikace pro automatické značkování objektů na fotografiích**
Kategorie: Uživatelská rozhraní
Akademický rok: 2024/25

Zadání:

1. Nastudujte způsoby ukládání značek do metadat souborů a nástroje pro detekci objektů v obrazu.
2. Navrhněte aplikaci, která pomocí existujících nástrojů rozpozná na fotografiích objekty a jejich popis zapíše do metadat souboru.
3. Integrujte vedoucím práce vybraný detektor objektů s webovou aplikací. Rozpoznávejte alespoň 30 různých typů objektů v obrázku.
4. Proveďte experimenty a otestujte správnost zapsaných a zobrazených dat.

Literatura:

- Li, S., Dragičević, S., Veenendaal, B.: Advances in Web-based GIS, Mapping Services and Applications, CRC Press, 2011, ISBN 978-0-415-89081-6
- Viayaraghavan, J., Dhanapal, Y.: ArcGIS for JavaScript Developers by Example. Packt Publishing, 2016, ISBN 978-1-78588-866-3
- MAREŠ, Jiří. Doplnění geografických souřadnic do fotografií podle záznamu trasy. Bakalářská práce na VUT FIT, 2018.

Při obhajobě semestrální části projektu je požadováno:
První dva body zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Semerád Lukáš, Ing., Ph.D.**
Vedoucí ústavu: Kočí Radek, Ing., Ph.D.
Datum zadání: 1.11.2024
Termín pro odevzdání: 14.5.2025
Datum schválení: 4.11.2024

Abstrakt

Táto bakalárska práca sa zaoberá vývojom webovej aplikácie na platforme Django, ktorá umožní automatické označovanie objektov na fotografiách s využitím modelov YOLO, ResNet a SSD. Súčasťou práce bude analýza a porovnanie uvedených metód, pričom cieľom bude určiť, ktorý z modelov je najvhodnejší na detekciu konkrétnych typov objektov. Aplikácia poskytne používateľom intuitívne užívateľské rozhranie, kde bude možné nahrávať fotografie a pridať automaticky vygenerované značky objektov priamo do metadát obrázkov. Implementácia bude prebiehať v programovacom jazyku Python s využitím frameworku Django pre serverovú časť. Cieľom projektu je vytvoriť funkčnú, presnú a efektívnu aplikáciu na spracovanie a označovanie fotografií.

Abstract

This bachelor's thesis focuses on the development of a web application on the Django platform for automatic object tagging in photographs using YOLO, ResNet, and SSD models. The project includes an analysis and comparison of these methods to determine which model is most suitable for detecting specific types of objects. The application will provide users with an intuitive interface, allowing them to upload photos and automatically generate object tags directly in the metadata of the images. The implementation will be carried out in Python, utilizing the Django framework for the backend. The goal of the project is to create a functional, accurate, and efficient application for processing and tagging photographs.

Klíčová slova

web, fotografia, metadáta, automatické označovanie objektov, YOLO, ResNet, SSD, Faster R-CNN, Python, Django, strojové učenie, EXIF, analýza modelov, spracovanie fotografií, počítačové videnie, neurónové siete

Keywords

web, photography, metadata, automatic object tagging, YOLO, ResNet, SSD, Faster R-CNN, Python, Django, machine learning, EXIF, model analysis, photo processing, computer vision, neural networks

Citace

FRÜŠTÖK, Patrik. *Aplikace pro správu fotek a editaci metadat*. Brno, 2024. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Lukáš Semerád, Ph.D.

Aplikace pro správu fotek a editaci metadat

Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Lukáša Semeráda, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....

Patrik Frůštůk

1. května 2025

Poděkování

Rád by som podakoval vedúcemu práce, pánovi Ing. Lukášovi Semerádovi, Ph.D. za užitočné rady, odborné vedenie a neustálu podporu.

Obsah

1	Úvod	4
2	Metadáta	5
2.1	Metadáta vo fotkách	5
2.2	Konkrétne typy metadát	5
2.3	Existujúce riešenia na správu a úpravu metadát vo fotkách	8
2.4	Bezpečnosť a súkromie metadát	10
3	Rozpoznávanie objektov v obrazoch	11
3.1	Úvod	11
3.2	Neuronové siete a ich učenie	11
3.3	Princípy konvolučných neurónových sietí (CNN)	12
3.4	Moderné metódy detekcie objektov	14
3.5	YOLO (You Only Look Once)	15
3.6	SSD (Single Shot Detector)	18
3.7	Faster R-CNN	20
3.8	Porovnanie modelov	22
4	Základ a návrh aplikácie	23
4.1	Základ aplikácie	23
4.2	Použitý framework Django	23
4.3	Návrh aplikácie	24
5	Analýza modelov	26
5.1	Presnosť detekcie	27
5.2	Rýchlosť detekcie	36
5.3	Vplyv prahovej hodnoty dôveryhodnosti	41
5.4	Záverečné porovnanie modelov	48
6	Záver	49
	Literatura	51

Seznam obrázků

2.1	Pôvodná fotografia.	7
2.2	Editovaná fotografia.	7
2.3	ExifTool GUI.	8
2.4	Príklad zobrazenia metadát v aplikácii Lightroom.	9
2.5	Značkovanie obrázkov v aplikácii Lightroom.	9
2.6	Rozhranie Darktable.	10
3.1	Princíp posilňovaného učenia.	12
3.2	Princíp konvolúcie.	13
3.3	Princíp pooling.	14
3.4	Princíp YOLO modelu.	16
3.5	Anchor boxy.	17
3.6	Architektúra SSD (Single Shot Detector).	19
3.7	Architektúra Faster R-CNN.	20
3.8	Architektúra ResNet.	21
3.9	Reziduálny blok v ResNet.	21
4.1	Môj návrh galérie.	25
5.1	Presnosť podľa veľkosti objektu – YOLO.	27
5.2	Presnosť podľa veľkosti objektu – YOLO.	28
5.3	Presnosť podľa počtu objektov – YOLO	29
5.4	Presnosť podľa počtu objektov – YOLO.	30
5.5	Presnosť podľa veľkosti objektu – Faster R-CNN.	30
5.6	Presnosť podľa veľkosti objektu – Faster R-CNN.	31
5.7	Presnosť podľa počtu objektov – Faster R-CNN.	32
5.8	Presnosť podľa počtu objektov – Faster R-CNN.	33
5.9	Presnosť podľa veľkosti objektu – SSD.	33
5.10	Úspešnosť detekcie malých objektov – SSD.	34
5.11	Presnosť podľa počtu objektov – SSD.	35
5.12	Kombinovaná analýza – SSD.	35
5.13	Distribúcia času inferencie – YOLO (vzorka).	36
5.14	Základné výkonnostné metriky – YOLO.	37
5.15	Distribúcia času inferencie – Faster R-CNN.	38
5.16	Výkonnostné metriky – Faster R-CNN.	38
5.17	Distribúcia času inferencie – SSD.	39
5.18	Výkonnostné metriky – SSD.	40
5.19	Hodnotiace metriky podľa prahovej hodnoty dôveryhodnosti.	41
5.20	Počty detekcií podľa prahovej hodnoty dôveryhodnosti.	42

5.21	Priemerná dôveryhodnosť podľa prahovej hodnoty.	42
5.22	Hodnotiace metriky podľa prahovej hodnoty dôveryhodnosti.	43
5.23	Počty detekcií podľa prahovej hodnoty dôveryhodnosti.	44
5.24	Priemerná dôveryhodnosť podľa prahovej hodnoty.	44
5.25	Hodnotiace metriky podľa prahovej hodnoty dôveryhodnosti.	45
5.26	Počty detekcií podľa prahovej hodnoty dôveryhodnosti.	46
5.27	Priemerná dôveryhodnosť podľa prahovej hodnoty.	46

Kapitola 1

Úvod

Predstavte si, že máte stovky fotografií z dovolenky. Niektoré zachytávajú nádherné pláže, iné historické pamiatky či jedinečné momenty s rodinou. Koľkokrát ste si už priali, aby ste mohli tieto fotografie automaticky triediť podľa toho, čo na nich je? Aby ste jedným kliknutím mohli zobrazíť všetky fotografie s vašim obľúbeným psom, všetky snímky západu slnka alebo fotografie, kde sú konkrétne objekty, ako napríklad autá či stromy? Moderné technológie nám dnes už umožňujú riešiť tieto úlohy pomerne rýchlo a presne – a to vďaka neurónovým sieťam.

Neurónové siete, inšpirované fungovaním ľudského mozgu, sú jednou z najpokročilejších oblastí umelej inteligencie. Ich schopnosť učiť sa z dát a aplikovať naučené poznatky na nové situácie nachádza využitie v mnohých oblastiach nášho života. Pomáhajú nám pri rozpoznávaní reči, pri preklade textov do cudzích jazykov, pri diagnostike chorôb v medicíne a dokonca aj pri autonómnom riadení vozidiel. V oblasti spracovania obrazu a fotografií sa stali nástrojom na detekciu objektov, rozpoznávanie tvárí či analýzu obsahu.

Ako nám však môžu neurónové siete pomôcť v každodennom živote? Predstavme si, že chcete triediť fotografie podľa obsahu, ale manuálne prechádzanie stoviek snímok by zabralo hodiny. Neurónové siete dokážu tento proces automatizovať. Modely ako YOLO (You Only Look Once), Faster R-CNN a SSD (Single Shot Multibox Detector) sú navrhnuté na rýchlu a presnú detekciu objektov na fotografiách. Predstavte si aplikáciu, ktorá by dokázala nielen rozpoznať objekty na vašich fotografiách, ale ich aj automaticky označiť a zaradiť do kategórií. Už nikdy by ste nemuseli hľadať konkrétnu fotografiu v nekonečných priečinkoch.

Táto práca sa zameriava na vývoj takejto aplikácie, ktorá kombinuje moderné technológie neurónových sietí a framework Django. Okrem samotnej implementácie bude skúmané, ktorý z modelov YOLO, Faster R-CNN a SSD je najvhodnejší na detekciu konkrétnych typov objektov. Cieľom je nielen vytvoriť funkčný nástroj, ale aj prispieť k lepšiemu pochopeniu toho, ako neurónové siete môžu zjednodušiť a zefektívniť každodenné činnosti, akou je napríklad správa a organizácia fotografií.

Kapitola 2

Metadáta

Metadáta alebo metaúdaje sú štrukturované dáta, ktoré obsahujú informácie o iných dátach. Ako tradičný príklad sa považuje katalogizačný lístok v knižnici. V tomto lístku sa nachádzajú dáta, ktoré uľahčujú vyhľadávanie kníh podľa pôvodu alebo umiestenia knihy. Tieto "metadáta" môžu obsahovať informácie o počte strán knihy, rozmeroch obrázkov, informácie o kontexte, meno autora...

Metadáta majú využitie vo viacerých odvetviach. Na webe sa napríklad využívajú na vyhľadávanie vo veľkom množstve informácií (napr. informácie v záhlaviach HTML dokumentov). **Metadáta vo zvukových súboroch** obsahujú informácie o názve skladby, autorovi skladby a pod.[22]

2.1 Metadáta vo fotkách

Metadáta vo fotografiách popisujú jej **technické parametre a podmienky, za akých bola daná fotografia odfotená**. Môžu obsahovať údaje o rozlíšení, formáte, veľkosti súboru, autorstve, právach a lokalizačné údaje. Umožňujú, bez potreby otvárania danej fotografie, zistiť o nej užitočné informácie.

Tri najbežnejšie typy metadát používané vo fotografiách sú **technické metadáta, právne metadáta a deskriptívne metadáta**. Technické metadáta obsahujú informácie o technických aspektoch procesu vytvárania fotografie, ako typ zariadenia, použitú clonu, čas uzávierky, ISO, ohniskovú vzdialenosť a softvér použitý na spracovanie fotografie. Právne metadáta zahŕňajú meno autora alebo tvorca fotografie, informácie o autorských právach a licenciách, práva na použitie modelov a majetku zobrazeného na fotografii, a podmienky licencovania a obmedzenia použitia fotografie. Deskriptívne metadáta poskytujú informácie o obsahu fotografie, vrátane názvu alebo popisu, kľúčových slov alebo tagov, autora alebo vlastníka fotografie, a dátumu a času, kedy bola fotografia zhotovená.[15]

Tieto informácie sú užitočné pri organizovaní fotografií a zohrávajú kľúčovú úlohu pri ich archivácii, správe a publikovaní. Napríklad pri práci s veľkými kolekciami fotografií umožňujú metadáta efektívne filtrovať obrázky podľa dátumu, miesta snímania či použitého fotoaparátu.

2.2 Konkrétne typy metadát

Existujú rôzne typy metadát používané v digitálnych fotografiách, z nich najbežnejšie sú EXIF, IPTC, XMP a Geotagging.

2.2.1 EXIF

EXIF (Exchangeable Image File Format) je najpoužívanejší štandard metadát, ktorý automaticky pridáva technické informácie o snímke priamo do obrazového súboru. Bol navrhnutý japonským priemyselným združením JEIDA. Exif využíva prakticky každý výrobca fotoaparátov. Chýba tu však podpora filmových formátov a obrázkov s vyššou farebnou hĺbkou. [2]

Exif dáta v sebe ukladajú rôzne informácie ako počet exponovaných snímkov fotoaparátom; značku, typ a model fotoaparátu; čas a dátum, kedy bola fotografia odfotená; a nastavenia fotoaparátu vrátane clony, rýchlosti uzávierky, ohniskovej vzdialenosti, ISO, použitia blesku, režimu fotografovania a orientácie obrázka. Obsahujú tiež náhľad obrázka (10 kB) použitý na zobrazenie na LCD displeji fotoaparátu alebo v programe na úpravu fotografií, informácie o polohe prevzaté napríklad z GPS snímača, a opis obrázka, autorské práva, komentár fotografujúceho a podobne. [9]

2.2.2 IPTC

IPTC-NAA-Standard bol definovaný zhruba v roku 1990, pôvodne ako Information Interchange Model (IIM) pre novinárske účely. Tento štandard slúži podobne ako EXIF k uloženiu textových informácií do obrazových súborov. Sú to dáta ako **autorské práva, meno autora, titulok fotografie a kľúčové slová pre vyhľadávanie**. Využívajú ho najmä spoločnosti, ktoré spravujú veľké množstvo fotografií (spravodajské spoločnosti, fotobanky...) a pomocou špecializovaného softvéru sa im táto správa zjednoduší a vyhľadávanie prebehne takmer okamžite. [15]

2.2.3 XMP

XMP (Extensible Metadata Platform) je formát metadát vytvorený spoločnosťou Adobe Systems, ktorý slúži na štandardizáciu metadát. Tento formát zahŕňa informácie z EXIF, IPTC a ďalších formátov a umožňuje ich ľahký prístup a správu. XMP sa líši od predchádzajúcich formátov tým, že obsahuje aj informácie o úpravách vykonaných na súbore RAW, čo znamená, že používateľ môže kedykoľvek vrátiť obrázok do pôvodného stavu, ak je to potrebné.

Pri práci so súbormi typu RAW je k nemu priložený XMP súbor, ktorý uchováva všetky informácie o posledných úpravách vykonaných v bitmapovom alebo vektorovom grafickom editore. Napríklad, ak bola na obrázku vykonaná zmena farebnej teploty pomocou Kelvinovej stupnice z pôvodnej hodnoty 4900 K na novú hodnotu 2000 K, editor automaticky vytvorí nový XMP súbor, v ktorom je možné vidieť túto zmenu.

XMP metadáta môžu byť vložené priamo do pôvodného súboru alebo uložené samostatne v bočnom súbore (sidecar). XMP metadáta zostávajú prítomné v súbore aj v prípade, že sa formát súboru zmení, napríklad z Photoshop súboru (PSD) na komprimovaný obrázok (JPEG). [10]

Obrázok 2.1 slúži ako pôvodná hodnota spolu s jeho súborom a hodnotou farebnej teploty 4900 K (riadok označený "###"). Na obrázku 2.2 a jeho kóde vidíme efekt po zmenení danej hodnoty na hodnotu 2000 K.



Obrázek 2.1: Pôvodná fotografia.

```
<zdf:Description>
  <xdf:Description rdf:about="">
    <crs:RawFileName>IMG_5668.CR2</crs:RawFileName>
    <crs:Version>6.0</crs:Version>
    <crs:ProcessVersion>5.7</crs:ProcessVersion>
    <crs:WhiteBalance>As Shot</crs:WhiteBalance>
    ##<crs:Temperature>4900</crs:Temperature>
    <crs:Shadows>5</crs:Shadows>
    <crs:Brightness>+50</crs:Brightness>
  </xdf:Description>
</zdf:Description>
```



Obrázek 2.2: Editovaná fotografia.

```
<zdf:Description>
  <xdf:Description rdf:about="">
    <crs:RawFileName>IMG_5668.CR2</crs:RawFileName>
    <crs:Version>6.0</crs:Version>
    <crs:ProcessVersion>5.7</crs:ProcessVersion>
    <crs:WhiteBalance>As Shot</crs:WhiteBalance>
    ##<crs:Temperature>2000</crs:Temperature>
    <crs:Shadows>5</crs:Shadows>
    <crs:Brightness>+50</crs:Brightness>
  </xdf:Description>
</zdf:Description>
```

2.2.4 Geotagging

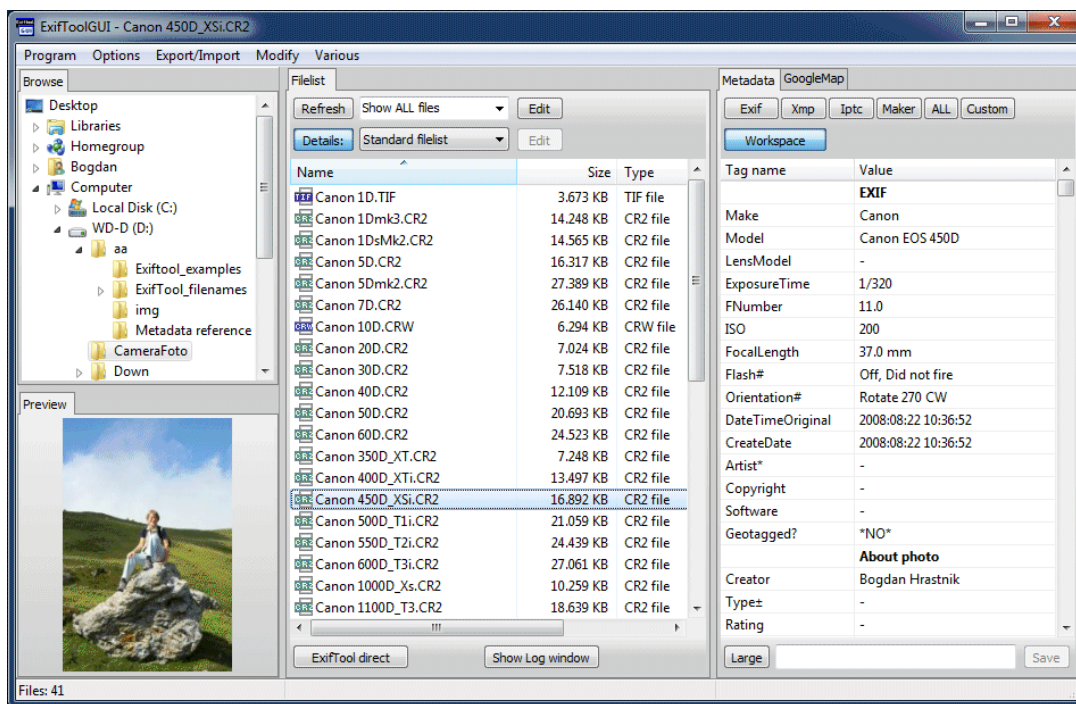
Geotagging alebo geocoding, je označenie pre pridávanie geografických metadát ku rôznym médiám (webové stránky, RSS zdroje, obrázky...). Prevažne ide o dáta ako **zemepisná dĺžka a šírka, nadmorská výška, azimut a zemepisné názvy**. [10]

2.3 Existujúce riešenia na správu a úpravu metadát vo fotkách

Možnosť spravovať a upravovať metadáta je kľúčová pri práci s veľkým objemom digitálnych snímok. Existuje množstvo nástrojov, ktoré umožňujú pridávať, upravovať a odstraňovať informácie (metadáta) v súboroch s fotografiami. Medzi najpoužívanejšie patrí ExifTool, Adobe Lightroom/Photoshop a Darktable.

2.3.1 ExifTool

ExifTool je špecializovaný nástroj, ktorý umožňuje podrobnú správu EXIF metadát vrátane úpravy geografických údajov či pridávania vlastných atribútov. Celkovo má však dosť zastaralé a neprehľadné grafické spracovanie, ako môžeme vidieť na obrázku 2.3. [12]

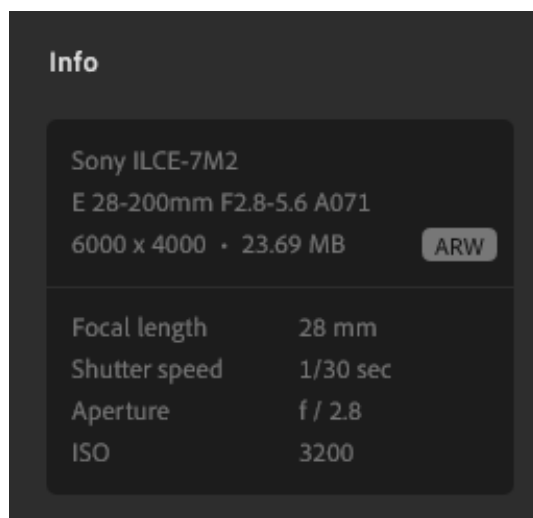


Obrázek 2.3: ExifTool GUI.

2.3.2 Adobe Lightroom

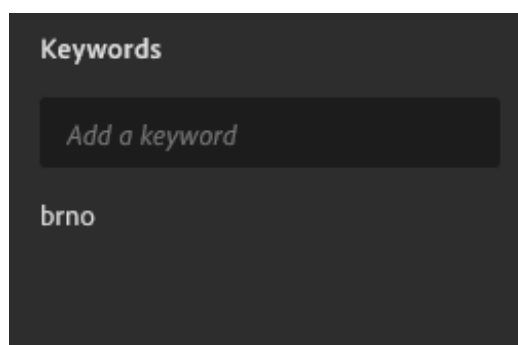
Adobe Lightroom ponúka intuitívne rozhranie na spravovanie IPTC, XMP a EXIF metadát viz. obr. 2.4. Umožňuje zobrazenie a ich úpravu, ktorá zahŕňa všetky časti úplného pracovného postupu fotografa, ktorý si tak vo väčšine prípadov vystačí len s touto aplikáciou. Typická výhoda a vlastnosť tejto aplikácie je takzvaný **katalog**. Katalog slúži ako vlastná

databáza fotografií, kde sú všetky uložené. Práve tento katalog drží v sebe informácie, ako sú metadáta o fotke.



Obrázek 2.4: Príklad zobrazenia metadát v aplikácii Lightroom.

Lightroom okrem iného ponúka aj pridanie značky objektu k obrázku viz. obr. 2.5. Nakoľko však táto funkcionality nie je jednou z hlavných tejto aplikácie, je to dosť zjednodušené, nevyužíva to žiaden typ automatickej detekcie a slúži to prevažne na manuálne triedenie fotografií užívateľom.



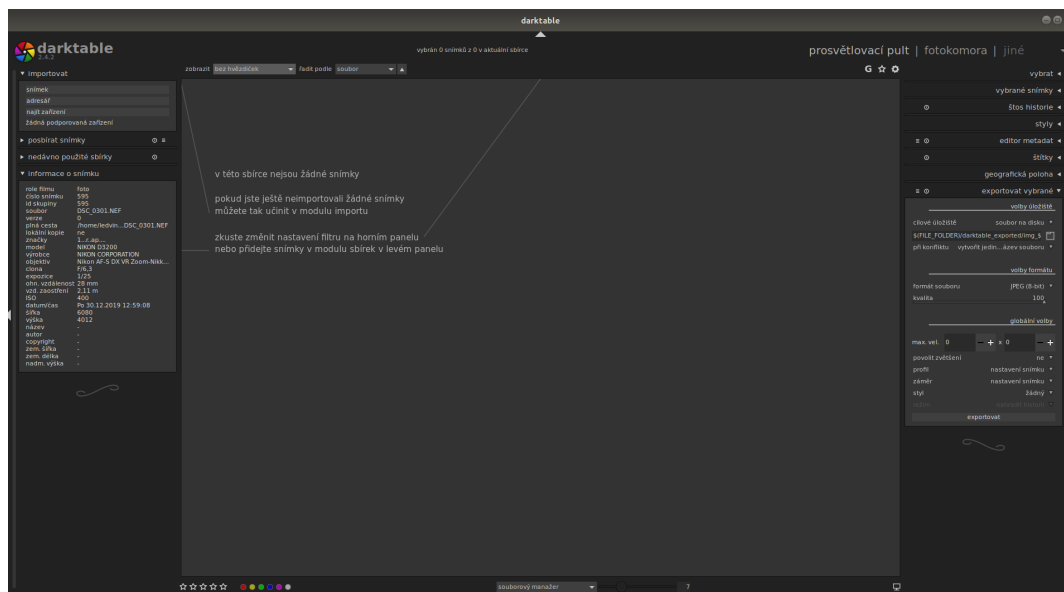
Obrázek 2.5: Značkovanie obrázkov v aplikácii Lightroom.

[1]

2.3.3 Darktable

Darktable je otvorený, voľne dostupný software na úpravu digitálnych fotografií a vývoj nezpracovaného grafického formátu raw. Je považovaný za alternatívu ku známejšiemu Lightroom a je obľúbený najmä medzi užívateľmi operačného systému Linux. Zaoberá sa nedeštruktívnou úpravou fotografií vo formáte raw.

Oproti iným aplikáciám, Darktable umožňuje svojim užívateľom používať na ovládanie svoje vlastné skripty v jazyku Lua. Na obrázku 2.6 je možné vidieť rozhranie tejto aplikácie.[7]



Obrázek 2.6: Rozhranie Darktable.

2.4 Bezpečnosť a súkromie metadát

Napriek tomu, že sú metadáta nesmierne užitočné a uľahčujú prácu fotografom a iným, nesú so sebou aj veľké množstvo rizík spojených primárne so súkromím.

Jedným z nich sú už spomínané geografické súradnice, ktoré sú súčasťou EXIF dát napríklad aj po pridaní fotky na internet. Môžu teda odhaliť adresu alebo aktuálnu polohu. Informácie ako model telefónu, softvér ktorým bola fotka upravená, prípadne sériové číslo vášho zariadenia, môžu byť zneužitú na sledovanie vašej aktivity online alebo využitie a napadnutie zraniteľných miest zariadenia.

Existuje viacero spôsobov, ako sa chrániť. Vypnutie geolokačných služieb zariadenia alebo vymazanie metadát pomocou špecializovaných aplikácií slúži ako istá forma prevencie.

Kapitola 3

Rozpoznávanie objektov v obrazoch

3.1 Úvod

Rozpoznávanie objektov v obrazoch je kľúčovou oblasťou počítačového videnia a umelej inteligencie. Táto disciplína sa zameriava na umožnenie strojom analyzovať a porozumieť obrazovým dátam podobne ako ľudské oko. Medzi hlavné úlohy rozpoznávania objektov patrí identifikácia objektov (detekcia prítomnosti konkrétnych objektov v obraze, napríklad identifikácia áut, ľudí alebo zvierat), klasifikácia podľa kategórií (priradenie objektu do príslušnej kategórie ako auto, bicykel, pes na základe naučených vlastností) a segmentácia scény (rozdelenie obrazu na jednotlivé oblasti tak, aby každý pixel patril konkrétnemu objektu alebo triede).

3.2 Neuronové siete a ich učenie

Neuronové siete sú kľúčovým nástrojom v oblasti rozpoznávania objektov v obrazoch. Zohrávajú dôležitú úlohu pri identifikácii, klasifikácii a segmentácii objektov. Tieto siete sa dokážu učiť a rozpoznať vzory a vlastnosti objektov v obrázkoch. Proces učenia je založený na prispôbovaní váh synapsií medzi neurónmi v sieti. [26]

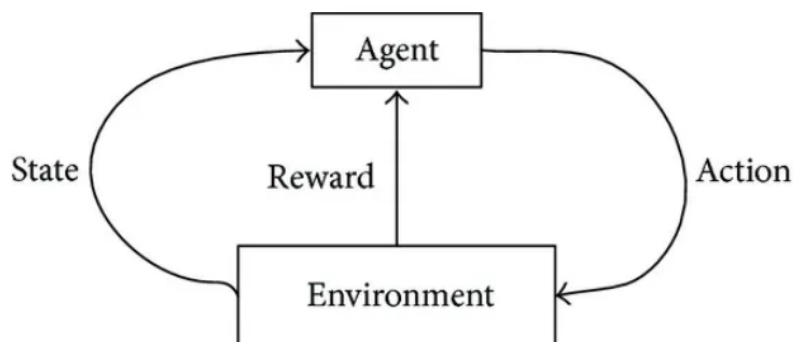
Arthur Samuel, pionier v oblasti strojového učenia, definoval tento proces ako oblasť, ktorá umožňuje počítačom učiť sa bez explicitného naprogramovania.

V kontexte rozpoznávania objektov, neurónové siete **analyzujú obrazové dáta** (napríklad pixely) a na základe vzorov, ktoré sa naučili z tréningových dát, dokážu presne **identifikovať objekty v nových obrázkoch**. Tento proces zahŕňa viacero krokov ako extrahovanie relevantných vlastností obrazu, porovnávanie s naučenými vzormi a priradovanie týchto vzorov k správnym triedam. [26]

Existujú rôzne **typy učenia**, ktoré sa používajú pri tréňovaní neurónových sietí. **Učenie s učiteľom (supervised learning)** je najbežnejší prístup pri rozpoznávaní objektov. Sieť dostáva človekom zadané tréningové dáta obsahujúce vstupy (obrázky) a ich správne už označené výstupy (označené objekty). Cieľom je, aby sieť našla vzory v obrazoch a priradovála objekty k príslušným kategóriám, ako je **auto, pes, alebo strom**. [3]

Učenie bez učiteľa (unsupervised learning) je menej bežný prístup v rozpoznávaní objektov, no môže byť užitočný pri analýze veľkého množstva neoznačených obrazových dát. Cieľom je, aby sieť sama identifikovala štruktúry v dátach, ako je zhľukovanie podobných objektov. Tento prístup sa môže využiť napríklad na zhľukovanie obrazov podľa vizuálnych podobností, bez nutnosti predchádzajúceho označenia objektov. [6]

Posilňované učenie predstavuje metódu strojového učenia, ktorá umožňuje tzv. agentovi, teda algoritmu alebo systému, zlepšovať svoje rozhodovanie na základe interakcie s prostredím. Cieľom tejto techniky je naučiť agenta vykonávať akcie, ktoré maximalizujú získanú odmenu alebo minimalizujú riziko. Na obrázku 3.1, je znázornený diagram princípu fungovania posilňovaného učenia.



Obrázek 3.1: Princíp posilňovaného učenia.

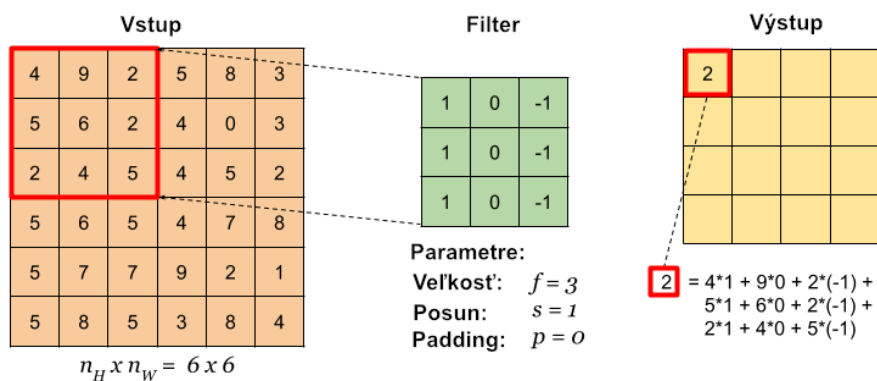
Proces učenia prebieha **iteratívne** – agent **pozoruje rôzne stavy prostredia, vykonáva akcie** a na základe spätnej väzby sa postupne **zdokonaľuje**. Táto metóda patrí medzi techniky umelej inteligencie a často sa modeluje pomocou Markovovho rozhodovacieho procesu, kde agent vyberá najvhodnejšiu akciu podľa aktuálneho stavu. Posilňovanie je zabezpečené prostredníctvom tzv. posilňovacieho signálu, ktorý slúži ako spätná väzba pre učenie.

Základné kroky posilňovaného učenia zahŕňajú: **pozorovanie aktuálneho stavu prostredia agentom; zvolenie a vykonanie konkrétnej akcie na základe rozhodovacieho mechanizmu; reakciu prostredia poskytnutím odmeny alebo posilňovacieho signálu; a uloženie informácií o stave, akcii a spätnej väzbe pre ďalšie učenie**. Takýmto spôsobom agent postupne zlepšuje svoje rozhodovanie a prispôbuje sa rôznym situáciám, čím dosahuje optimálny výkon.[24]

3.3 Princípy konvolučných neurónových sietí (CNN)

Konvolučné neurónové siete (CNN) sú základom moderného rozpoznávania objektov a sú špeciálne navrhnuté na spracovanie obrazových dát. Ich architektúra je postavená na vrstvenom usporiadaní, ktoré umožňuje efektívne extrahovať a analyzovať charakteristické črty obrazu. Kľúčové komponenty CNN zahŕňajú konvolučné vrstvy, pooling vrstvy a plne prepojené vrstvy.

Konvolučné vrstvy vykonávajú konvolúciu medzi vstupnými dátami a filtermi (jadrami), ktoré identifikujú špecifické vlastnosti obrazu, ako sú hrany, textúry alebo tvary. Princíp konvolúcie spočíva v aplikovaní jadra (filtrovej matice) na lokálne časti obrazu, čím sa vytvára mapa vlastností. Tieto mapy vlastností zachytávajú rôzne vizuálne vzory, ktoré umožňujú CNN identifikovať dôležité črty obrazu. Na figure 3.2 je znázornený princíp konvolúcie. [14]

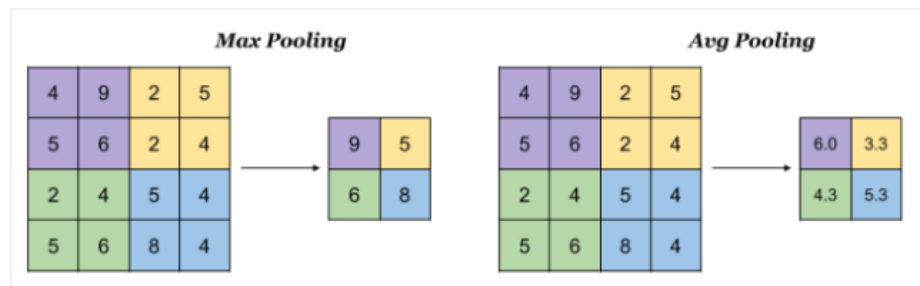


Obrázek 3.2: Princíp konvolúcie.

Medzi základné vlastnosti **konvolúcie** patrí **lineárnosť** (konvolúcia je lineárnou operáciou, čo znamená, že výsledok operácie zachováva základnú štruktúru kombinovaných vlastností), **posunová invariantnosť** (výsledok konvolúcie nezávisí od toho, kde sa na obraze nachádzajú prvky, ktoré filter zachytáva) a **kombinácia vlastností** (umožňuje kombinovať rôzne charakteristiky, ako sú kontrasty a gradienty). [19]

Najdôležitejšie **konvolučné masky** zahŕňajú **Sobelovu masku** (používa sa na detekciu hrán v obraze tým, že identifikuje oblasti s vysokým gradientom), **Laplacián** (slúži na zvýraznenie hrán a kontúr prostredníctvom druhého derivátu) a **Gaussian Blur** (používa sa na redukcii šumu a vyhladenie obrazu). [17]

Pooling vrstvy sú navrhnuté na redukcii priestorových rozmerov dát, čo znižuje výpočtovú náročnosť a zároveň zjednodušuje výstupy. Okrem toho pooling vrstvy zvyšujú odolnosť CNN voči drobným zmenám v obraze, ako sú posuny, šum alebo rotácie.[30] Medzi najbežnejšie typy pooling vrstiev patria **Max pooling** (vyberá najvyššiu hodnotu z každého okna vstupného obrazu, používa sa častejšie, pretože zachováva najvýznamnejšie rysy obrazu a robí výsledné charakteristiky robustnejšími) a **Avg pooling** (počíta priemernú hodnotu z každého okna, je menej častý, ale vhodný pre aplikácie, kde sú dôležité priemerné vlastnosti a trendy, nie iba maximálne hodnoty). [14] Princíp max a avg pooling je zobrazený na figure 3.3.



Obrázek 3.3: Princíp pooling.

Plne prepojené vrstvy slúžia na kombináciu extrahovaných vlastností z predchádzajúcich vrstiev a umožňujú konečnú klasifikáciu alebo predikciu hodnôt. Tieto vrstvy fungujú ako tradičné neurónové siete, kde sú všetky uzly prepojené, čo umožňuje vytvoriť komplexné reprezentácie extrahovaných črt. [30]

Tento hierarchický proces, kde nižšie vrstvy detekujú základné vizuálne črty (napr. hrany) a vyššie vrstvy kombinujú tieto črty do komplexnejších vzorov (napr. identifikácia objektov), je kľúčom k vysokej presnosti CNN. Vloženie operácií, ako je normalizácia, pooling alebo dropout, navyše zvyšuje efektivitu a generalizačné schopnosti siete, čím sa CNN stávajú ideálnymi pre široké spektrum aplikácií v počítačovom videní.

3.4 Moderné metódy detekcie objektov

V súčasnosti existujú dva hlavné prístupy k detekcii objektov. **Jednostupňové metódy**, ako YOLO (You Only Look Once) a SSD (Single Shot Detector), vykonávajú detekciu a klasifikáciu v jednom kroku, pričom si zachovávajú dostatočnú presnosť. To ich robí rýchlymi a vhodnými pre aplikácie v reálnom čase. Na druhej strane, **dvojstupňové metódy** delia proces detekcie na dve fázy. Najprv generujú oblasti záujmu, kde by sa mohli nachádzať

objekty, a následne ich klasifikujú. Tento prístup, ako napríklad Faster R-CNN, zvyšuje presnosť detekcie, ale je časovo náročnejší.

3.5 YOLO (You Only Look Once)

YOLO (You Only Look Once) je jedným z najznámejších a najpoužívanějších modelov na detekciu objektov v reálnom čase. Tento model sa stal veľmi populárnym v oblasti počítačového videnia vďaka svojej rýchlosti a efektívnosti. Jeho základný princíp spočíva v tom, že celý obraz sa analyzuje jednorazovo, čo znamená, že predikcia objektov sa vykonáva počas jediného prechodu obrazom cez model. [23]

3.5.1 Základný princíp YOLO

Model YOLO rozdelí vstupný obraz na mriežku o rozmeroch $S \times S$, kde každá bunka tejto mriežky je zodpovedná za predikciu objektov, ktoré sa nachádzajú v tejto oblasti. Každá bunka mriežky predikuje určité hodnoty:

- x a y : sú súradnice stredu ohraničujúceho rámčeka v rámci danej bunky mriežky. Tieto hodnoty sú normalizované v rámci rozmerov bunky, takže sa pohybujú v rozsahu od 0 do 1.
- w a h : sú šírka a výška ohraničujúceho rámčeka. Tieto hodnoty sú tiež normalizované, aby model nebol závislý od rozlíšenia obrázkov.
- s : je istota detekcie objektu v danej bunke. Hodnota vyjadruje pravdepodobnosť, že sa v tejto bunke nachádza objekt (medzi 0 a 1).
- $c_1, c_2, \dots, c_n$: sú pravdepodobnosti pre jednotlivé triedy objektov, ktoré model môže detegovať (napr. osoba, pes, auto). [13]

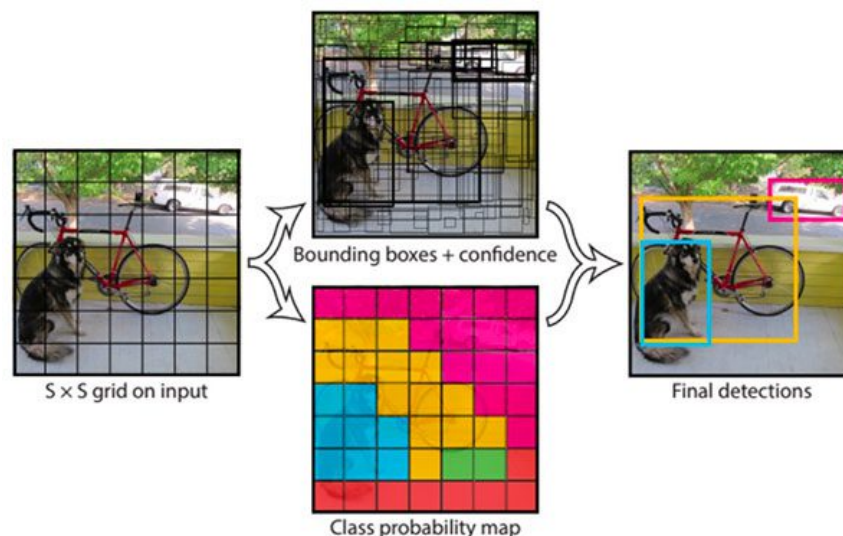
Výstupom modelu je teda množina predikcií, ktoré sa kombinujú na určenie ohraničujúcich rámčiekov a príslušných tried objektov. Základný princíp fungovania spolu s $S \times S$ mriežkou, mapou pravdepodobnosti a ohraničujúcimi rámčekmi sú znázornené na obrázku 3.4.

3.5.2 Rýchlosť a efektívnosť YOLO

Jednou z najväčších výhod modelu YOLO je jeho rýchlosť. Tradičné metódy detekcie objektov, ako sú RCNN (Regions with Convolutional Neural Networks), vyžadujú viacero prechodov cez rôzne oblasti obrázka. Naopak, YOLO vykonáva detekciu objektov v jednom kroku, čo umožňuje spracovanie obrázkov v reálnom čase. Tento prístup je ideálny pre aplikácie, ako sú autonómne vozidlá, monitorovanie bezpečnosti či rozpoznávanie tvárí v reálnom čase. [13]

3.5.3 Vývoj YOLO a jeho verzie

Model YOLO (You Only Look Once) prešiel od svojho vzniku v roku 2016 významným vývojom, pričom každá nová verzia priniesla zlepšenia v presnosti, rýchlosti a efektívnosti detekcie objektov.

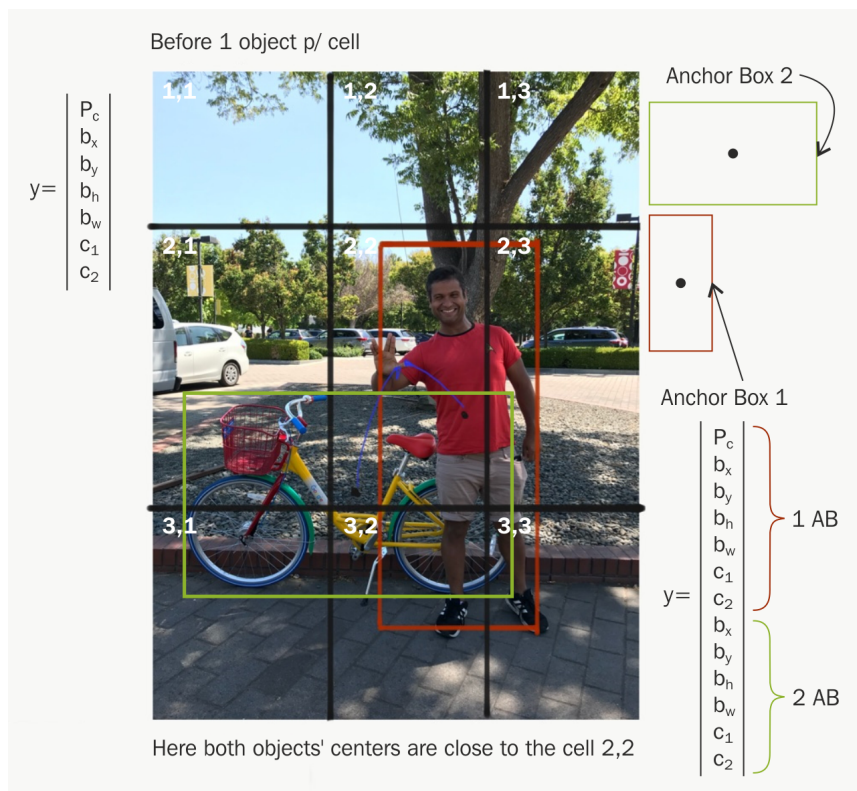


Obrázek 3.4: Princíp YOLO modelu.

YOLOv1 Prvá verzia YOLO bola predstavená v roku 2016. Tento model spracovával celý obraz naraz a vytváral predikcie o objektoch a ich umiestnení v jednom kroku. Tento prístup bol revolučný, pretože eliminoval potrebu extrakcie kandidátskych oblastí (regions of interest) a umožnil detekciu objektov v reálnom čase. **YOLOv2 (YOLO9000)** zaviedol techniky ako **batch normalization** a **anchor boxy**, čím zlepšil schopnosť modelu pracovať s rôznymi veľkosťami objektov. Táto verzia tiež umožnila tréning na kombinovaných **datasetoch**, čím zvýšila počet tried, ktoré model dokáže detegovať. **YOLOv3** priniesol zlepšenia v detekcii menších objektov a umožnil predikcie pre viaceré triedy naraz. Použil viacúrovňovú architektúru na predikciu objektov na rôznych škálach, čím zlepšil presnosť detekcie. **YOLOv4** sa zamerala na optimalizáciu výkonu modelu. Zaviedla techniky ako **CSPNet (Cross Stage Partial Networks)** a **augmentácie dát**, ktoré zlepšili presnosť a rýchlosť detekcie. **YOLOv5**, vyvinutý komunitou, sa zameria na jednoduchosť implementácie a tréningu modelu. Použil ľahšie a rýchlejšie architektúry, čím sa stal populárnym pre rôzne praktické aplikácie. **YOLOv6 a YOLOv7** priniesli ďalšie vylepšenia v architektúre siete a optimalizácii výkonu. **YOLOv8** zaviedol pokročilé techniky hlbokého učenia, čím dosiahol lepšiu presnosť a efektívnosť. Vylepšenia zahŕňali optimalizáciu stratových funkcií a efektívnejšie využitie výpočtových zdrojov. **YOLOv9**, predstavený v roku 2024, priniesol techniky ako **Programmable Gradient Information (PGI)** a **Generalized Efficient Layer Aggregation Network (GELAN)**. Tieto inovácie výrazne zlepšili efektívnosť, presnosť a adaptabilitu modelu. **YOLOv10** uvedený v máji 2024, dosiahol nižšiu latenciu a menší počet parametrov v porovnaní s predchádzajúcimi verziami, čím sa stal špičkou v detekcii objektov v reálnom čase. **YOLOv11** predstavený v decembri 2024, priniesol významné zlepšenia v oblasti adaptívneho učenia a detekcie objektov v extrémnych podmienkach. Model implementuje novú architektúru **Dynamic Neural Routing (DNR)**, ktorá umožňuje flexibilné spracovanie rôznych typov obrazových vstupov. Táto verzia je optimalizovaná pre nasadenie na zariadeniach s nízkou výpočtovou kapacitou a zároveň poskytuje špičkový výkon v reálnom čase. [28]

3.5.4 Pokročilé techniky v YOLO

Anchor Boxes sú preddefinované rámčeky rôznych veľkostí a pomerov strán, ktoré sa používajú pri detekcii objektov na obraze. Tieto rámčeky sú umiestnené na každý pixel alebo bod mriežky v obraze a slúžia ako predpoklady pre možnú polohu objektu. Každý rámček sa hodnotí podľa toho, ako dobre sa zhoduje s objektmi v obraze. Princíp fungovania je znázornený na obrázku 3.5 Tento prístup zlepšuje predikciu objektov rôznych veľkostí, pretože model nemusí začať s predpokladmi o veľkosti objektu, ale môže sa prispôsobiť rôznym rozmerom rámčekov. Tento prístup umožňuje efektívnejšie a presnejšie detegovanie objektov, najmä v prípadoch, keď objekty majú rôzne veľkosti a tvary. [4]



Obrázek 3.5: Anchor boxy.

Batch Normalization je technika, ktorá normalizuje výstupy jednotlivých vrstiev v neurónových sieťach. Cieľom je znížiť vnútorné kovariátne posunutie (internal covariate shift), ktoré sa objavuje počas tréningu, keď sa distribúcia aktivácií medzi vrstvami mení. Normalizácia zabezpečuje, že výstupy každej vrstvy majú strednú hodnotu 0 a rozptyl 1. Tým sa zlepšuje stabilita modelu a zrýchľuje tréning, pretože umožňuje použitie vyšších učebných rýchlostí a znižuje potrebu použitia pravidelných techník ako Dropout. [25]

Focal Loss je stratová funkcia navrhnutá na zlepšenie výkonu modelov pri riešení problémov s triednou nevyváženosťou, ktoré sa vyskytujú v úlohách detekcie objektov. V tradičných stratových funkciách, ako je "krížová entropia" (meria rozdiel medzi skutočnými a predikovanými pravdepodobnosťami tried), sa modely zameriavajú rovnako na všetky triedy. Avšak pri detekcii objektov, kde môžu byť niektoré objekty veľmi malé alebo ťažko detekovateľné, môže byť veľká časť tréningových dát menej dôležitá. Focal Loss zameriava po-

zornosť modelu na ťažšie prípady, ktoré sú menej zastúpené, čím zlepšuje presnosť detekcie pri malých objektoch alebo pri objektoch, ktoré sú často zamieňané s pozadím.[27]

IoU (Intersection over Union) je metrika používaná na hodnotenie presnosti predikcií v úlohách detekcie objektov. Meria sa ako podiel plochy priesečníka medzi predikovaným rámčekom a skutočným rámčekom objektu k ploche ich zjednotenia. Tento ukazovateľ sa často používa na hodnotenie kvality detekcie a lokalizácie objektu. Vysoká hodnota IoU znamená, že predikovaný rámček sa veľmi dobre zhoduje so skutočným rámčekom objektu, čo indikuje presnú lokalizáciu. Pri hodnotení modelov detekcie objektov sa často používa prahová hodnota IoU, aby sa určil úspech predikcie. Model je považovaný za úspešný, ak IoU prekročí určitú prahovú hodnotu, zvyčajne 0,5. IoU je kľúčovou metrikou pre modely ako YOLO, SSD a Faster R-CNN. [20]

3.5.5 Využitie YOLO v praxi

YOLO sa široko používa v rôznych oblastiach: **Autonómne vozidlá** (Detekcia chodcov, vozidiel a prekážok). **Bezpečnostné kamery** (Monitorovanie a analýza videa v reálnom čase). **Medicínske aplikácie** (Detekcia patologických nálezov na lekárskech snímkach). **Sociálne médiá a zábava** (Analýza videí a obrázkov).

3.5.6 Výzvy a obmedzenia

Aj keď je YOLO veľmi rýchly a efektívny, má určité obmedzenia. Detekcia veľmi malých objektov je menej presná, pretože sa obrázok rozdeľuje na mriežku. Problémy môžu nastať aj pri detekcii objektov v prekrývajúcich sa oblastiach.

3.6 SSD (Single Shot Detector)

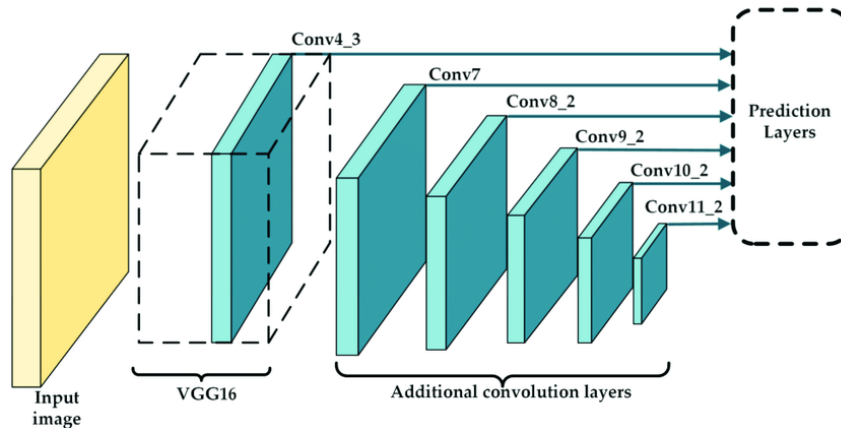
Model SSD (Single Shot Detector) je populárny prístup pre detekciu objektov v reálnom čase. Na rozdiel od modelu YOLO (You Only Look Once) využíva viacero vrstiev konvolučnej siete, ktoré pracujú s príznakmi v rôznych mierkach. Táto vlastnosť umožňuje SSD lepšie detegovať objekty rôznych veľkostí, čím sa odlišuje od YOLO, ktoré používa jednotnú mriežku na predikciu hraníc objektov.

3.6.1 Architektúra SSD

Znázornená na obrázku 3.6 pozostáva z dvoch hlavných komponentov:

Základná sieť (backbone): Typicky ide o predtrénovanú konvolučnú sieť, ako je VGG16 alebo ResNet, ktorá slúži na extrakciu príznakov z obrazu. Na rozdiel od YOLO, ktoré používa jednotnú štruktúru siete, SSD využíva viacero vrstiev na rôznych úvratiach, čím poskytuje viac detailov o objektoch rôznych mierok.

Konvolučné detekčné hlavy: Na základnú sieť sú pripojené dodatočné konvolučné vrstvy, ktoré generujú mapy príznakov s rôznymi rozlíšeniami. Každá vrstva deteguje objekty v konkrétnej mierke, čo zlepšuje výkon pre menšie aj väčšie objekty. YOLO, na druhej strane, využíva jednu mriežku, čo môže obmedzovať presnosť pri detekcii malých objektov. [16]



Obrázek 3.6: Architektúra SSD (Single Shot Detector).

3.6.2 Mechanizmus detekcie

SSD využíva 'default' alebo 'anchor' boxy, ktoré sú priradené k bunkám v mapách príznakov. Tieto boxy sú preddefinované a pokrývajú rôzne veľkosti a pomery strán, čo umožňuje detekciu objektov rôznych veľkostí v rôznych miestach obrazu. Pre každú bunku v mriežke, ktorá je vytvorená na základe výstupu základnej siete, SSD predpovedá dve hlavné hodnoty:

Klasifikačné skóre: Pre každý anchor box model predpovedá pravdepodobnosť, že box obsahuje objekt určitej triedy. Tieto skóre sa potom používajú na určenie, ktoré boxy obsahujú relevantné objekty.

Regresné posuny: SSD tiež predpovedá korekcie pre každý anchor box, ktoré umožňujú, aby sa boxy presnejšie prispôbili skutočným hraniciam objektov. Tieto regresné posuny sa používajú na "jemné doladenie" pozície a veľkosti boxov, čím sa zvyšuje presnosť detekcie.

Výhodou SSD je, že každá vrstva konvolučnej siete v architektúre deteguje objekty **v rôznych mierkach**. Týmto spôsobom model dokáže zachytiť objekty rôznej veľkosti bez potreby zložitejších stratégií ako v prípade iných modelov, napríklad R-CNN. SSD tiež používa techniku **neskoršej nestruktúrovanej selekcie** (non-maximum suppression), ktorá umožňuje odstrániť redundantné predpovede a zachovať len najpresnejšie detekcie. [18]

Tento prístup je veľmi efektívny, pretože umožňuje vykonávať detekciu v reálnom čase pri zachovaní vysokej presnosti, čo je kľúčové pre aplikácie, ako je autonómna jazda alebo monitorovanie bezpečnosti.

3.6.3 Výhody a výkon SSD

SSD je ideálny pre aplikácie, kde je potrebné kombinovať presnosť a rýchlosť. SSD300 (s veľkosťou vstupu 300×300 pixelov) dosahuje porovnateľnú presnosť s YOLOv3, ale lepšie deteguje malé objekty. Na druhej strane, YOLO je stále rýchlejší a vhodnejší pre aplikácie, kde je prioritou reálny čas.

3.6.4 Aplikácie

SSD sa používa v oblastiach ako **autonómne vozidlá** (detekcia prekážok rôznych veľkostí), **video dohľad** (lepšia detekcia menších objektov v zložitých prostrediach), **mobilné aplikácie** (efektívne využitie na zariadeniach s obmedzeným výpočtovým výkonom).

3.7 Faster R-CNN

Faster R-CNN (Region-based Convolutional Neural Network) je pokročilý model pre detekciu objektov, ktorý predstavuje významný pokrok v oblasti strojového učenia a počítačového videnia. Tento model je navrhnutý na presnú identifikáciu a lokalizáciu objektov v obraze, pričom spája rýchlosť a presnosť. [21]

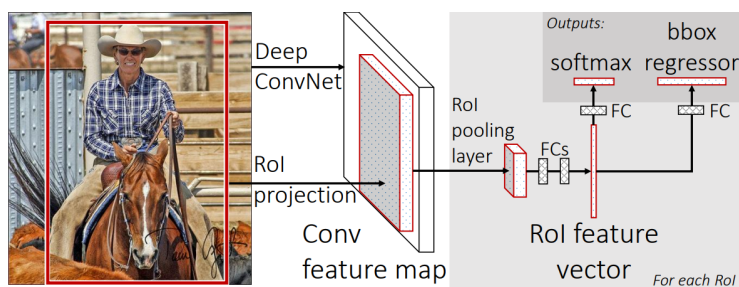
3.7.1 Čo je Faster R-CNN

Faster R-CNN je rozšírením modelov R-CNN a Fast R-CNN. V porovnaní s predchodcami rieši problém pomalého generovania návrhov oblastí (region proposals) pomocou Region Proposal Network (RPN). Táto sieť generuje návrhy oblastí priamo z obrazových dát, čím sa výrazne zvyšuje rýchlosť detekcie. [21]

3.7.2 Architektúra Faster R-CNN

Architektúra Faster R-CNN znázornená na obrázku 3.7 zahŕňa:

- **Konvolučná sieť:** Používa sa na extrakciu základných vlastností obrazu. Bežne sa využívajú siete ako ResNet alebo VGG.
- **Region Proposal Network (RPN):** Generuje návrhy oblastí s rôznou pravdepodobnosťou výskytu objektov.
- **ROI Pooling:** Normalizuje návrhy oblastí na jednotnú veľkosť, čo umožňuje ich spracovanie v klasifikátore.
- **Klasifikátor a regresor:** Klasifikátor identifikuje objekty a regresor upravuje hranice ohraničujúcich rámcov. [11]



Obrázek 3.7: Architektúra Faster R-CNN.

3.7.3 Prepojenie s ResNet

Faster R-CNN často využíva ResNet ako základnú konvolučnú sieť pre extrakciu vlastností z obrazu. ResNet, vďaka svojej architektúre založenej na reziduálnych blokoch, umožňuje efektívne tréňovanie veľmi hlbokých modelov bez problémov s miznúcimi gradientmi. Tento prístup zlepšuje presnosť a stabilitu detekcie objektov, najmä v prípadoch, kde je potrebné spracovať komplexné scény alebo malé objekty.[5]

Reziduálne bloky a skratové spojenia v ResNet

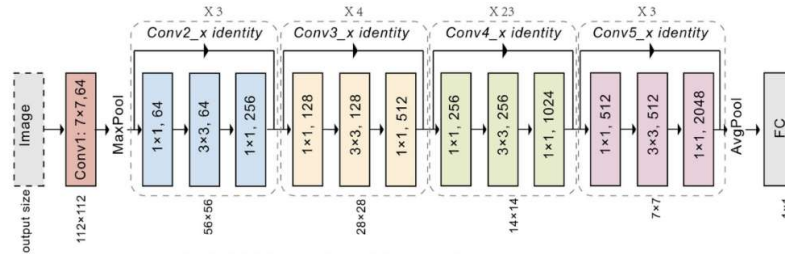
ResNet je neurónová sieť, ktorá rieši problém miznúcich gradientov pomocou tzv. reziduálnych blokov (*residual blocks*). Tieto bloky umožňujú, aby sa časti dát preskočili určité vrstvy a následne boli opäť začlenené do výpočtu. Tento prístup je užitočný pre veľmi hlboké siete, pričom kľúčovým prvkom sú tzv. skratové spojenia (angl. skip connections). Matematicky sa tento koncept vyjadruje ako:

$$H(x) = f(x) + x$$

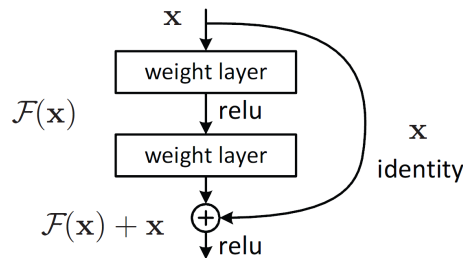
Kde $f(x)$ predstavuje transformáciu vstupu a x je pridaný vstup cez skratové spojenie. Tieto skratové spojenia umožňujú gradientu obísť problémové vrstvy, čím sa zlepšuje proces učenia a architektúra je robustnejšia voči šumu a nesprávnym vrstvám.[29]

Architektúra ResNet

Architektúra ResNet je znázornená na obrázku 3.8. Tento obrázok ilustruje použitie reziduálnych blokov, ktoré obsahujú dve konvolučné vrstvy a skratové spojenie. Reziduálne bloky umožňujú jednoduchšie učenie hlbokých sietí tým, že udržiavajú informácie z predchádzajúcich vrstiev. Obrázok 3.9 zobrazuje detailný pohľad na reziduálny blok, kde sa vstup pridáva k výstupu transformácie. [29]



Obrázek 3.8: Architektúra ResNet.



Obrázek 3.9: Reziduálny blok v ResNet.

Prínos ResNet v Faster R-CNN

Použitie ResNet ako základnej siete v Faster R-CNN zlepšuje schopnosť modelu extrahovať komplexné vlastnosti z obrazových dát. To je obzvlášť dôležité pri detekcii malých objektov alebo detailov, kde jednoduchšie architektúry často zlyhávajú. ResNet zároveň umožňuje, aby Faster R-CNN profitoval z hlbkej architektúry bez zvýšenia rizika saturácie presnosti alebo degradácie výkonu. [8]

3.8 Porovnanie modelov

Kľúčové vlastnosti popísaných algoritmov zhrňuje nasledujúca tabuľka:

Tabuľka 3.1: Porovnanie moderných modelov detekcie objektov.

Model	Rýchlosť	Presnosť	Typy objektov	Výhody a nevýhody
YOLO	Veľmi vysoká	Stredná až vysoká	Vhodné pre veľké a stredne veľké objekty, ako sú vozidlá, osoby alebo zvieratá	+ Vysoká rýchlosť, vhodné pre reálny čas; - Menej presné pri detekcii malých alebo prekrývajúcich sa objektov
SSD	Vysoká	Stredná	Dobré pre stredne veľké objekty, ako sú tváre, predmety v interiéri alebo jednoduché scény	+ Efektívne pre obmedzené zdroje, vhodné pre mobilné zariadenia; - Menej presné pri komplexných scénach
Faster R-CNN	Stredná až nízka	Vysoká	Výborné pre malé a detailné objekty, ako sú text, jemné vzory alebo špecifické detaily na obraze	+ Veľmi vysoká presnosť, schopnosť identifikovať zložité objekty; - Vyššia výpočtová náročnosť, pomalšie spracovanie

3.8.1 Prvotná analýza modelov

YOLO: Silná stránka tohto modelu je jeho schopnosť efektívne a rýchlo detegovať objekty, čo ho robí ideálnym pre aplikácie v reálnom čase, ako je monitorovanie dopravy alebo bezpečnostné systémy. Avšak, pri detekcii malých objektov alebo objektov v preplnených scénach môže byť jeho presnosť nižšia.

SSD: SSD je vhodný na detekciu objektov strednej veľkosti, pričom vyniká v aplikáciách, ktoré vyžadujú vyváženie medzi presnosťou a rýchlosťou. Jeho výkon však klesá pri komplexných scénach alebo pri detekcii veľmi malých objektov.

Faster R-CNN: Tento model exceluje pri detekcii malých a detailných objektoch. Je ideálny pre aplikácie, ktoré kladú dôraz na vysokú presnosť, avšak jeho vyššia výpočtová náročnosť môže byť nevýhodou pre systémy s obmedzenými zdrojmi.

Kapitola 4

Základ a návrh aplikácie

4.1 Základ aplikácie

V rámci mojej bakalárskej práce som sa rozhodol implementovať webovú aplikáciu, ktorá umožňuje užívateľom spravovať svoje galérie fotografií. Kľúčovou funkcionalitou mojej aplikácie, ako naznačuje názov mojej témy, je automatické označovanie objektov na fotografiách pomocou troch modelov na detekciu objektov: YOLO (You Only Look Once), SSD (Single Shot Multibox Detector) a Faster R-CNN (Region-Based Convolutional Neural Network). Tieto značky sa fotke pridávajú automaticky po nahraní fotografie do galérie. Je možné ich upraviť, a slúžia primárne na triedenie fotiek a rýchle nájdenie v galérii.

Každý z týchto modelov má svoje špecifické vlastnosti a oblasti využitia, preto som pred samotnou integráciou modelov do mojej stránky, previedol analýzu, počas ktorej som testoval, pravdepodobnosť správnosti detekcie jednotlivých modelov aspoň na 30-tich rôznych typoch objektov. Na základe týchto analýz sa užívateľovi v mojej aplikácii odporúča, na ktorý typ objektu je vhodné použiť dané modely detekcie.

4.2 Použitý framework Django

Django je open-source framework pre vývoj webových aplikácií v jazyku Python. Poskytuje množstvo nástrojov na rýchlu tvorbu a škálovanie webových aplikácií. Framework sa riadi princípom DRY (Don't Repeat Yourself), čo mi zaistilo efektívne využívanie kódu bez zbytočného opakovania rovnakých častí.

Medzi najvýznamnejšie vlastnosti Django frameworku, ktoré využívam vo svojej aplikácii, patrí:

- **ORM (Object-Relational Mapping):** Umožňuje mi intuitívnu prácu s databázou prostredníctvom Python objektov.
- **Bezpečnostné mechanizmy:** Robustná ochrana pred bežnými webovými hrozbami ako XSS alebo CSRF útokmi, čo bolo pre moju aplikáciu kľúčové.
- **Modularita:** Flexibilná možnosť rozširovania mojej aplikácie o dodatočné moduly podľa požiadaviek.

4.2.1 Ako funguje Django

Django implementuje princíp MVC (Model-View-Controller) architektúry, ktorý som využil nasledovne:

- **Model:** Zabezpečuje definíciu štruktúry mojej databázy a spravuje dáta.
- **View:** Implementuje moju aplikačnú logiku a spracúva užívateľské požiadavky.
- **Controller:** Riadi tok dát medzi modelom a zobrazením, zabezpečuje spracovanie požiadaviek a aplikáciu business logiky.

V Django sú Templates zodpovedné za generovanie HTML výstupu pre zobrazenie užívateľského rozhrania mojej aplikácie vo webovom prehliadači. Tieto šablóny mi umožňujú oddeliť prezentačnú vrstvu od logiky aplikácie.

V mojej aplikácii Django slúži aj ako backend framework, ktorý som využil na zabezpečenie komplexnej správy užívateľských účtov, galérie fotografií, spracovanie požiadaviek a integráciu sofistikovaných modelov na detekciu objektov.

4.2.2 Potrebné knižnice

Na implementáciu mojej aplikácie a plnohodnotnú integráciu modelov YOLO, SSD a Faster R-CNN som využil nasledujúce knižnice:

- Django: Robustný framework, ktorý tvorí základ mojej webovej aplikácie.
- Django REST Framework (DRF): Rozšírenie, ktoré mi umožnilo vytvárať API pre efektívnu komunikáciu medzi frontendom a backendom.
- Pillow: Komplexná knižnica, ktorú som využil na pokročilú prácu s obrazovými súbormi v Python.
- PyTorch: Výkonný framework, ktorý mi poskytol zázemie pre tréning a využívanie modelov YOLO, SSD a Faster R-CNN.
- Torchvision: Obsahuje širokú škálu implementácií predtrénovaných modelov a transformácií obrazov, ktoré som využil vo svojom riešení.
- Ultralytics: Špecializovaná knižnica, ktorá mi umožnila efektívnu prácu s modelmi YOLO.
- OpenCV: Poskytla mi nástroje na sofistikované spracovanie a zobrazenie obrazov.

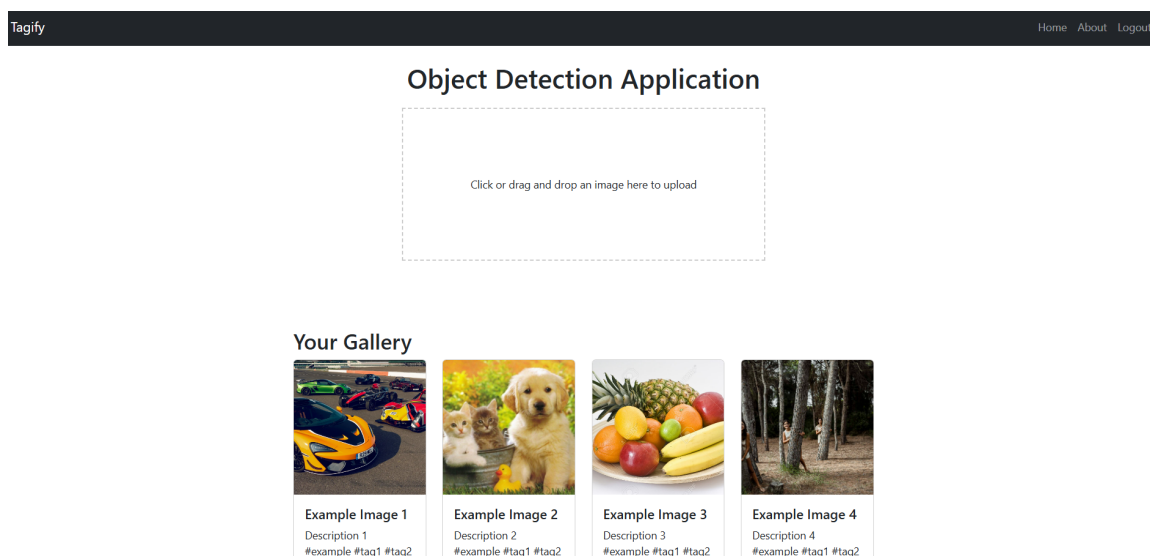
4.3 Návrh aplikácie

4.3.1 Funkcionalita aplikácie

Moja aplikácia poskytuje nasledujúce kľúčové funkcionality:

- Registrácia a prihlásenie užívateľov: Každý užívateľ disponuje v mojej aplikácii personalizovaným účtom a vlastnou galériou fotografií.
- Nahrávanie fotografií: Implementoval som intuitívnu možnosť nahrania fotografií do galérie.

- Vytváranie a upravovanie galérií.
- Automatické označovanie objektov: Vo svojom riešení využívam pokročilé modely YOLO, SSD a Faster R-CNN na presné rozpoznávanie a detekciu objektov na fotografiách.
- Filtrovanie a triedenie: S využitím nájdených značiek má užívateľ možnosť rýchlo a intuitívne nájsť hľadaný obrázok.
- Úprava metadát: Pri detekcii sa automaticky zapisujú nájdené značky do metadát fotografií.
- Štatistika analyzovaných modelov: Vytvoril som detailné porovnanie presnosti jednotlivých modelov.
- Odporúčanie modelu: Na základe mojej hĺbkovej analýzy je užívateľovi automaticky odporúčaný najvhodnejší model zodpovedajúci jeho špecifickým potrebám.



Obrázek 4.1: Môj návrh galérie.

Kapitola 5

Analýza modelov

Ako bolo uvedené v teoretickej časti tejto práce, jedným z cieľov je analyzovať a porovnať vybrané detekčné modely objektov – konkrétne **YOLO**, **SSD** a **Faster R-CNN**. Táto analýza predstavuje dôležitý krok pri výbere najvhodnejšieho modelu užívateľom pri značkovani a detekciu objektov na jeho fotkách.

Porovnanie je založené na viacerých kľúčových metrikách, ako sú:

- **presnosť detekcie** podľa veľkosti objektu, podľa počtu objektov na obrázku a podľa náročnosti scény,
- **počty detekcií** podľa prahovej hodnoty (**Threshold**),
- priemerná **dôveryhodnosť** podľa prahovej hodnoty,
- **rýchlosť spracovania** meraná vo FPS (*Frames Per Second*) a distribúcia času inferencie,
- výskyt **falošne pozitívnych (FP)** a **falošne negatívnych (FN)** detekcií,

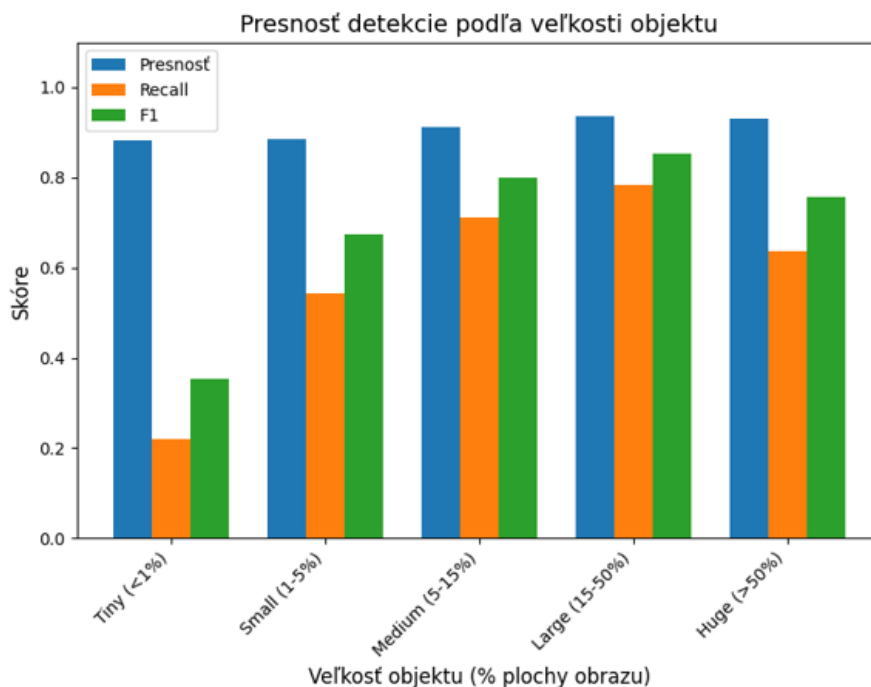
Cieľom tejto analýzy je identifikovať výhody a nevýhody jednotlivých prístupov, na základe ktorých bude možné odporučiť používateľom najvhodnejší model podľa ich potrieb – napríklad preferencia vyššej presnosti, rýchlosti spracovania alebo univerzálnosti detekcie naprieč rôznymi typmi objektov.

Nasledujúce podkapitoly prezentujú výsledky analýzy vo forme grafov a ich podrobný popis.

5.1 Presnosť detekcie

Presnosť detekcie je jednou z najdôležitejších metrík pri hodnotení výkonnosti detekčných modelov. V tejto podkapitole sú analyzované výsledky jednotlivých modelov z pohľadu presnosti na základe rôznych faktorov, ako sú veľkosť objektu, počet objektov na obrázku alebo náročnosť scény. V grafoch sú použité metriky ako "Presnosť" (Precision – podiel správne detegovaných objektov zo všetkých detegovaných objektov, t. j. schopnosť modelu generovať čo najmenej falošných poplachov), "Recall" (podiel správne detegovaných objektov zo všetkých existujúcich objektov v obraze, t. j. schopnosť modelu nájsť všetky objekty) a "F1 skóre" (harmonický priemer medzi presnosťou a recallom, ktorý poskytuje vyvážený pohľad na výkonnosť modelu).

Model YOLO



Obrázek 5.1: Presnosť podľa veľkosti objektu – YOLO.

Na obrázku 5.1 je znázornený graf, ktorý porovnáva tri základné metriky detekcie objektov — **Presnosť (Precision)**, **Úplnosť (Recall)** a **F1 skóre** — v závislosti od veľkosti objektu v obraze. Veľkosť objektu je vyjadrená ako percentuálny podiel plochy objektu vzhľadom na celkovú plochu obrazu a rozdelená do piatich kategórií:

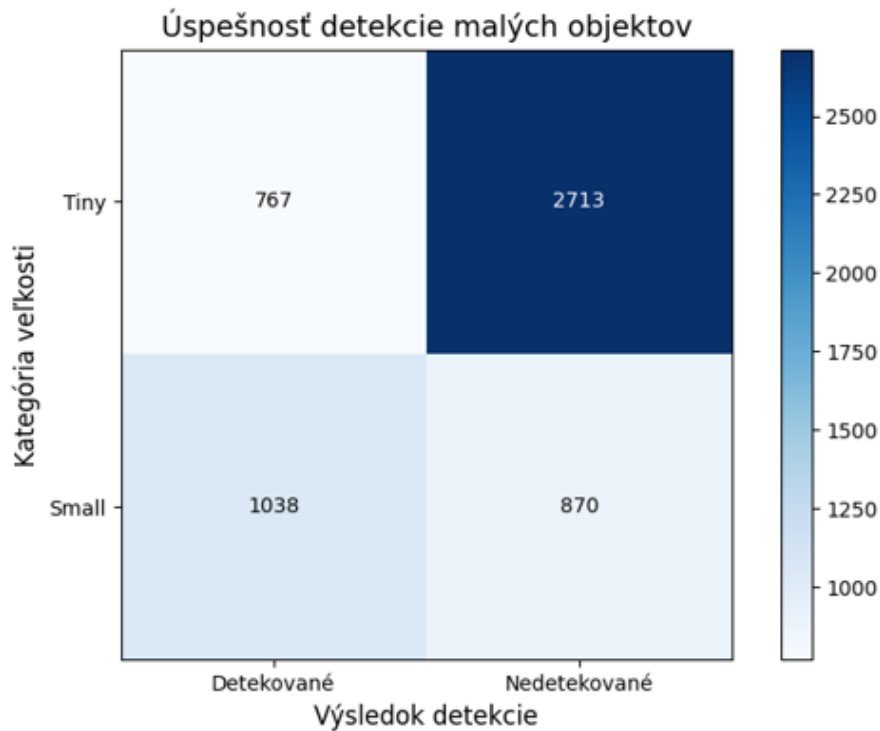
- **Tiny (<2%)** – veľmi malé objekty,
- **Small (1–5%)** – malé objekty,
- **Medium (5–15%)** – stredne veľké objekty,
- **Large (15–50%)** – veľké objekty,

- **Huge (>50%)** – veľmi veľké objekty.

Z grafu možno pozorovať, že **presnosť detekcie (Precision)** zostáva relatívne vysoká vo všetkých kategóriách objektov, pričom najnižšie hodnoty dosahuje pri kategóriách „Tiny“ a „Small“. Naopak, **úplnosť (Recall)** je výrazne nižšia pri detekcii veľmi malých objektov, čo poukazuje na to, že model nedokáže spoľahlivo zachytiť všetky výskyty týchto objektov v obraze.

F1 skóre, ktoré predstavuje harmonický priemer medzi presnosťou a úplnosťou, kopíruje trend oboch predchádzajúcich metrík a poskytuje vyvážený pohľad na efektivitu detekcie. Je zreteľné, že najnižšie F1 skóre je zaznamenané pri detekcii objektov kategórie „Tiny“, pričom s rastúcou veľkosťou objektu skóre postupne stúpa. Najlepšie výsledky sú dosiahnuté v kategóriách „Large“ a „Medium“, čo potvrdzuje, že detekčné modely majú vyššiu spoľahlivosť pri stredne veľkých až veľkých objektoch.

Pre detailnejší pohľad na správanie detekčného modelu pri malých objektoch bola vytvorená maticová vizualizácia znázornená na obrázku 5.2. Táto zobrazuje počet objektov v kategóriách *Tiny* a *Small* rozdelený podľa toho, či boli modelom úspešne detegované alebo nie.

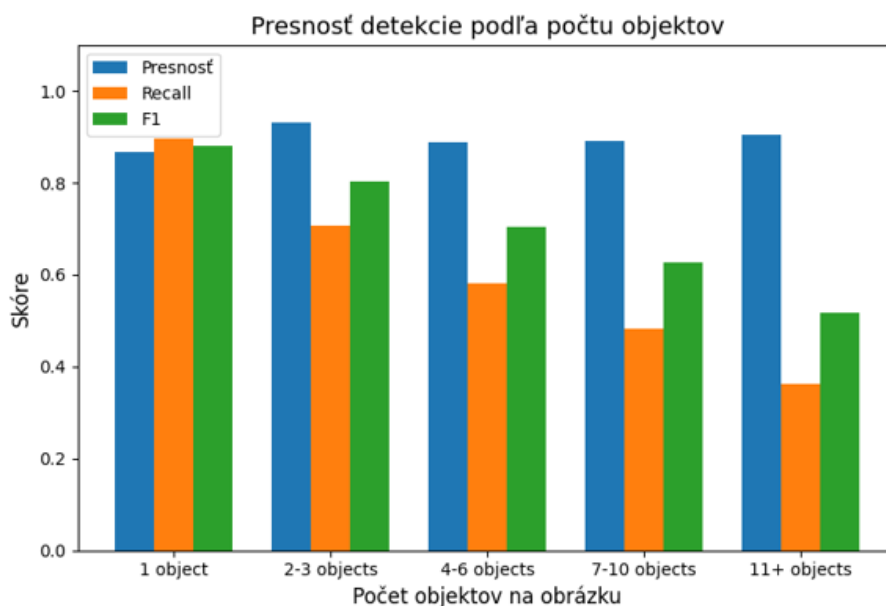


Obrázek 5.2: Presnosť podľa veľkosti objektu – YOLO.

Z tohto grafu vyplýva, že v kategórii *Tiny* bolo detegovaných **767** objektov, zatiaľ čo **2713** zostalo nedetegovaných. V prípade kategórie *Small* bolo detegovaných **1038** objektov a **870** nedetegovaných. Tieto výsledky poukazujú na vplyv veľkosti objektu na úspešnosť detekcie, tá je pri malých objektoch menej spoľahlivá, a model nie je schopný detekovať všetky malé objekty na obrázku.

Ďalším dôležitým faktorom, ktorý ovplyvňuje úspešnosť detekcie objektov, je počet objektov nachádzajúcich sa na jednej fotografii. Táto metrika je z pohľadu navrhovanej aplikácie pre správu fotografií a editáciu metadát rovnako kľúčová, keďže reálne fotografie často obsahujú viacero objektov rôznych typov. S rastúcim počtom objektov v zábere môže dôjsť k zvýšeniu náročnosti detekcie, a tým aj k zníženiu výkonnosti modelu.

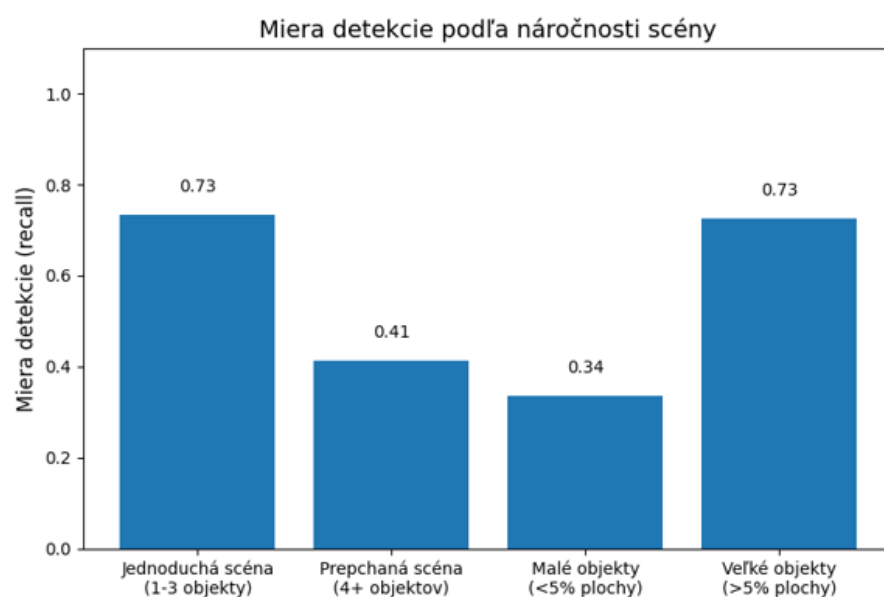
Graf na obrázku 5.3 ukazuje, ako sa mení presnosť, recall a F1 skóre v závislosti od počtu objektov na obrázku. Pri jednoduchých snímkach, kde sa nachádza iba jeden objekt, sú výsledky najlepšie – model dosahuje vysokú presnosť aj recall. S rastúcim počtom objektov však skóre postupne klesá, najmä v prípade recall. To znamená, že model častejšie „prehliada“ niektoré objekty, a teda ich vôbec nezachytí.



Obrázek 5.3: Presnosť podľa počtu objektov – YOLO

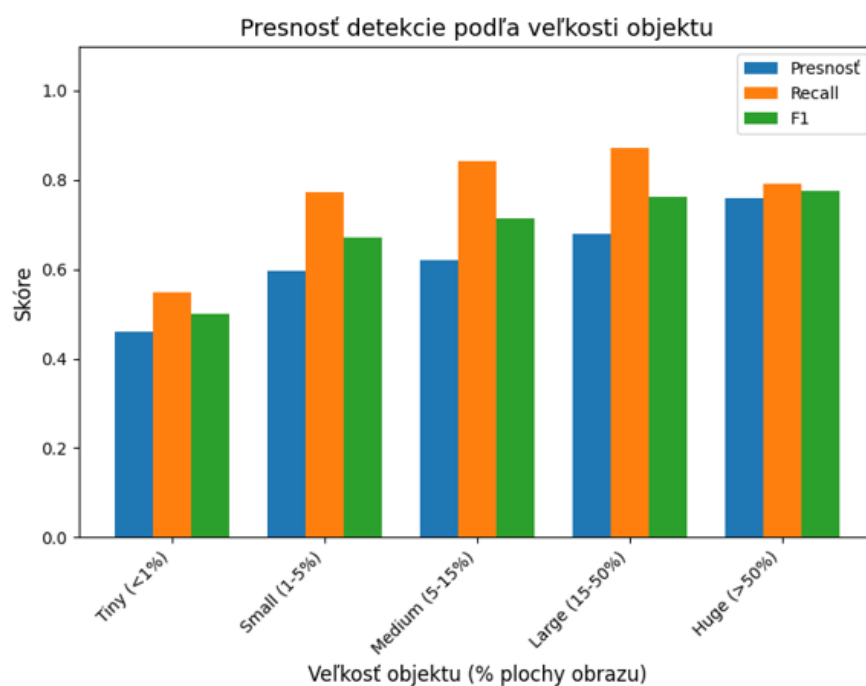
Príčin tohto poklesu môže byť viac. Ak je na snímke veľa objektov, môžu sa navzájom prekryvať, čo sťažuje ich správnu detekciu. Okrem toho, model má obmedzené výpočtové kapacity, a keď musí spracovať viacero objektov naraz, môže dôjsť k nepresnostiam v ich lokalizácii. Tento problém je najvýraznejší pri obrázkoch s 11 a viac objektmi, kde recall výrazne klesá.

Na poslednom grafe 5.4, som vybral 4 rôzne náročnosti scény. Výsledky len potvrdzujú, že najlepšiu detekciu dosahuje model pri jednoduchých scénach a pri obrázkoch s väčšími objektmi. V oboch prípadoch bola miera detekcie (recall) na úrovni 0,73. Naopak, pri preplnených scénach a malých objektoch model zjavne zaostáva – recall tu klesol na 0,41, resp. 0,34.



Obrázek 5.4: Presnosť podľa počtu objektov – YOLO.

Model Faster R-CNN



Obrázek 5.5: Presnosť podľa veľkosti objektu – Faster R-CNN.

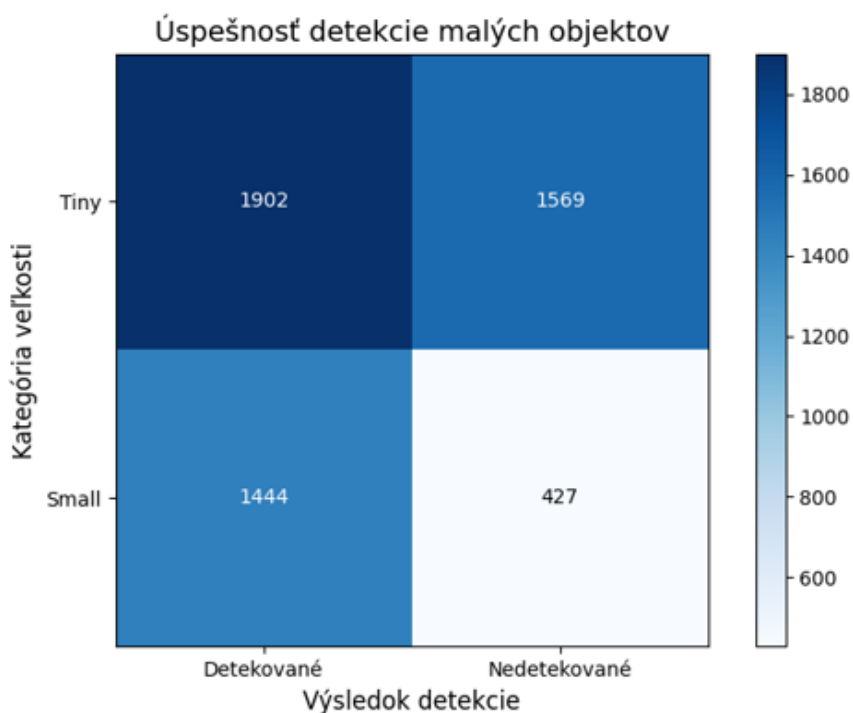
Na obrázku 5.5 je znázornený graf, ktorý rovnako ako pri modeli YOLO porovnáva metriky **Presnosť (Precision)**, **Úplnosť (Recall)** a **F1 skóre** v závislosti od veľkosti objektu v obraze. Kategórie veľkostí zostávajú rovnaké:

- **Tiny** (<2%) – veľmi malé objekty,
- **Small** (1–5%) – malé objekty,
- **Medium** (5–15%) – stredne veľké objekty,
- **Large** (15–50%) – veľké objekty,
- **Huge** (>50%) – veľmi veľké objekty.

Model Faster R-CNN dosahuje relatívne vysokú **presnosť (Precision)** vo všetkých kategóriách objektov, pričom mierne poklesy sú opäť zaznamenané pri menších objektoch – najmä v kategóriách „Tiny“ a „Small“. **Recall** je na tom však v porovnaní s YOLO o niečo lepšie, čo znamená, že model častejšie zachytí aj menšie objekty, ktoré YOLO prehliadol.

F1 skóre kopíruje trend oboch metrík a ukazuje lepšiu vyváženosť medzi presnosťou a recall pri menších objektoch, než tomu bolo pri YOLO. Výrazné zlepšenie F1 skóre možno pozorovať práve pri kategórii „Small“, čo poukazuje na vyššiu spoľahlivosť modelu Faster R-CNN pri menších objektoch.

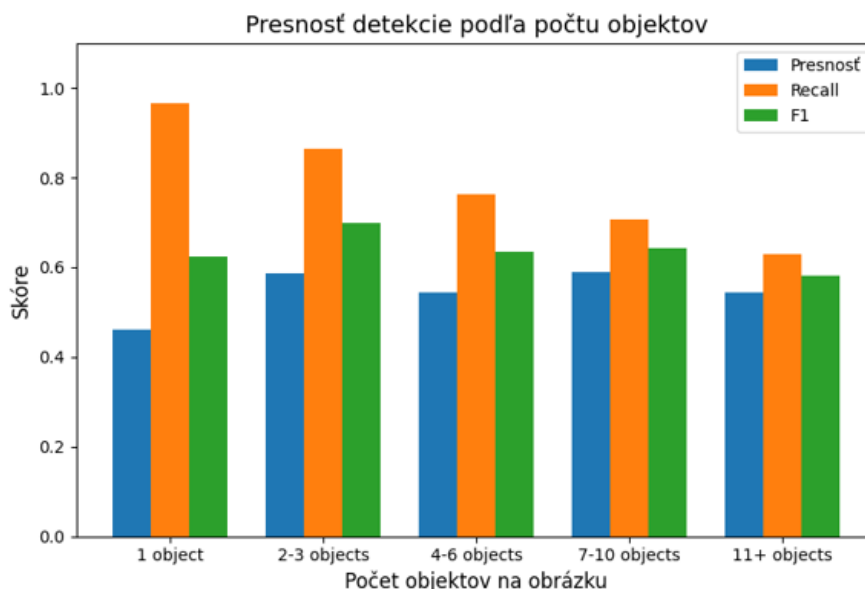
Na obrázku 5.6 je zobrazená vizualizácia zameraná na počet detegovaných a nedetegovaných objektov v kategóriách *Tiny* a *Small*. Podobne ako pri YOLO, táto analýza pomáha lepšie pochopiť správanie modelu pri detekcii malých objektov.



Obrázek 5.6: Presnosť podľa veľkosti objektu – Faster R-CNN.

Z grafu vyplýva, že v kategórii *Tiny* bolo detegovaných **1902** objektov, zatiaľ čo **1569** zostalo nedetegovaných. V prípade kategórie *Small* bolo detegovaných **1444** objektov a **427** nedetegovaných. V porovnaní s YOLO ide o lepšie výsledky, najmä v kategórii „Tiny“, kde Faster R-CNN dokázal detegovať viac objektov s vyššou presnosťou.

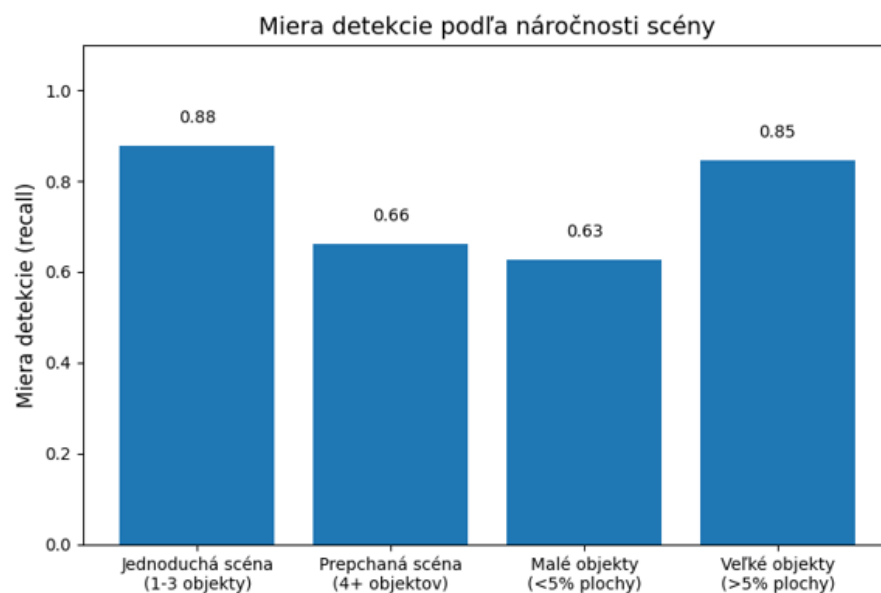
Ďalším analyzovaným faktorom je opäť počet objektov nachádzajúcich sa na jednej fotografii. Na obrázku 5.7 je znázornený graf zobrazujúci závislosť metriky Precision, Recall a F1 skóre od počtu objektov v obraze.



Obrázek 5.7: Presnosť podľa počtu objektov – Faster R-CNN.

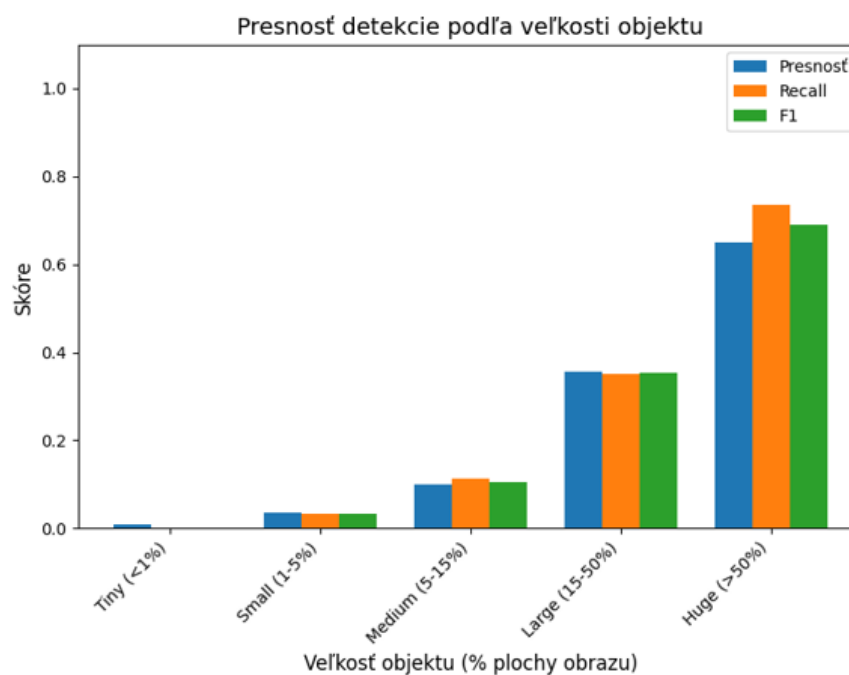
Podobne ako pri modeli YOLO, aj pri Faster R-CNN dochádza k poklesu výkonnosti pri rastúcom počte objektov v obraze. Kým pri 1–2 objektoch sú všetky metriky veľmi vysoké, pri preplnených scénach recall postupne klesá, aj keď pokles nie je až taký strmý ako pri YOLO. To naznačuje, že Faster R-CNN lepšie zvláda komplexnejšie scény s viacerými objektmi, ale stále naráža na limity pri veľmi zložitých záberoch.

Posledný graf 5.8 ukazuje kombinovanú analýzu vplyvu veľkosti objektov a zložitosti scény. Výsledky potvrdzujú, že najlepšie skóre dosahuje model pri väčších objektoch v jednoduchých scénach – recall dosahuje hodnotu až 0,78. V preplnených scénach s malými objektmi model opäť zaostáva, ale recall je mierne vyšší než v prípade YOLO – na úrovni 0,45, resp. 0,39.



Obrázek 5.8: Presnosť podľa počtu objektov – Faster R-CNN.

Model SSD

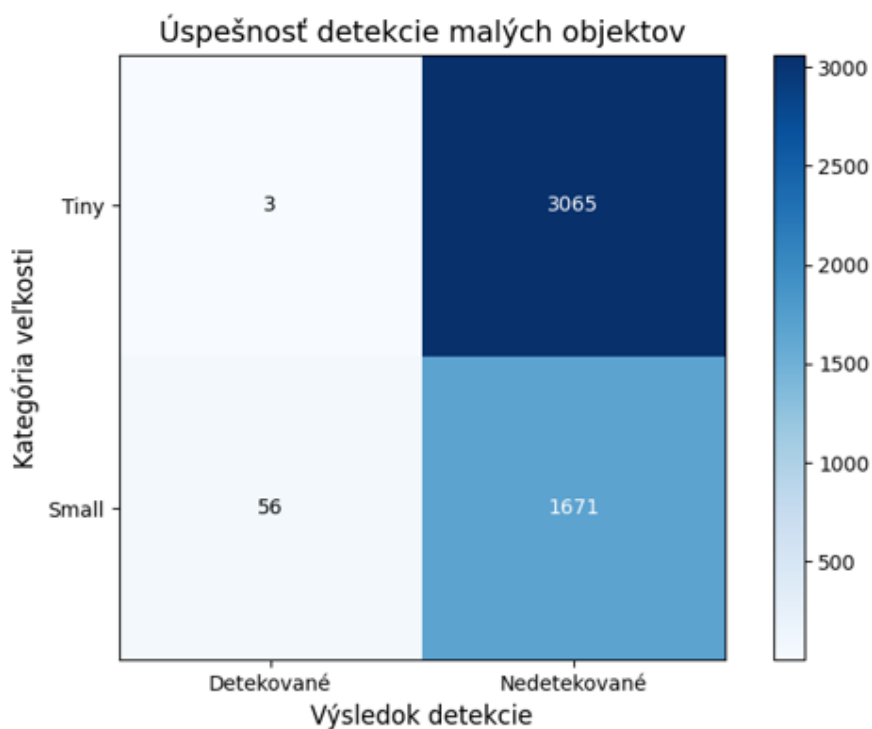


Obrázek 5.9: Presnosť podľa veľkosti objektu – SSD.

Na obrázku 5.9 je znázornená výkonnosť modelu SSD v závislosti od veľkosti objektov. V porovnaní s modelmi YOLO a Faster R-CNN je zreteľné, že SSD dosahuje najnižšie

hodnoty všetkých troch metrík – **Presnosť (Precision)**, **Úplnosť (Recall)** aj **F1 skóre**. Výrazný problém nastáva pri detekcii malých objektov, kde je recall extrémne nízky, čo znamená, že model väčšinu z nich vôbec nezachytí.

Kým YOLO si pri malých objektoch aspoň čiastočne zachováva použiteľnosť a Faster R-CNN je relatívne spoľahlivé, SSD v kategóriách „Tiny“ a „Small“ takmer úplne zlyháva. Dokonca aj pri väčších objektoch je jeho výkonnosť výrazne nižšia, čo naznačuje, že model sa horšie prispôsobuje rôznym veľkostiam objektov a trpí vyššou mierou falošných negatív.



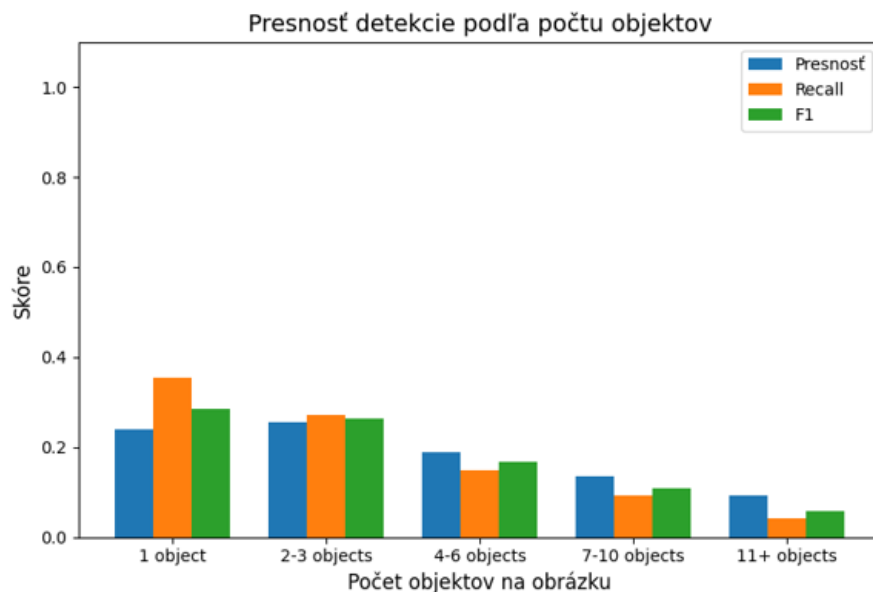
Obrázek 5.10: Úspešnosť detekcie malých objektov – SSD.

Graf na obrázku 5.10 ilustruje, aké veľké rozdiely sú medzi detegovanými a nedetegovanými objektmi v kategóriách „Tiny“ a „Small“. SSD je v tejto oblasti jednoznačne najhorší – v porovnaní s YOLO, ktorý dokázal detegovať aspoň časť malých objektov, SSD prakticky žiadne nezachytáva.

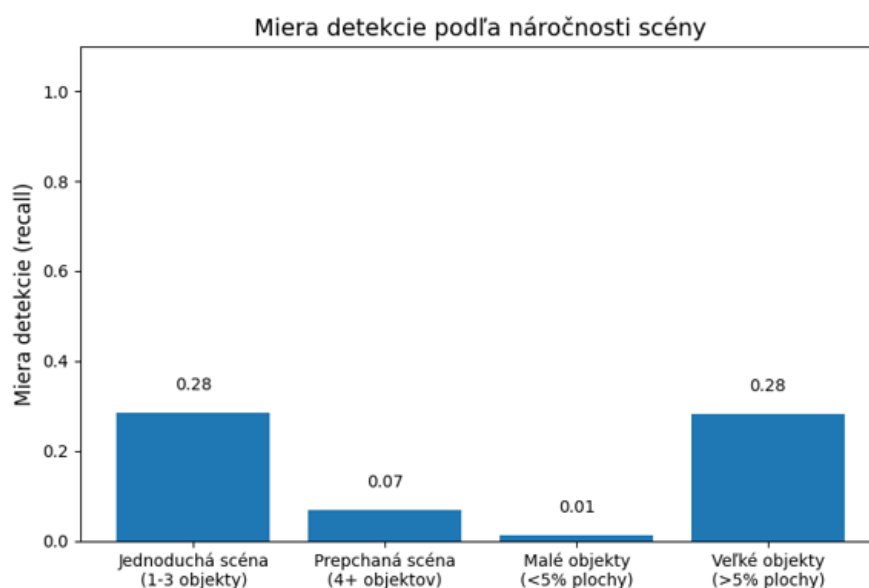
Ďalším faktorom ovplyvňujúcim výkonnosť je počet objektov v jednej scéne. Na obrázku 5.11 je znázornený pokles presnosti s narastajúcim počtom objektov.

Vidíme, že SSD má problém nielen s malými objektmi, ale aj s preplnenými scénami. Kým YOLO a Faster R-CNN vykazovali postupný pokles recallu so zvyšujúcim sa počtom objektov, SSD už pri 4 až 6 objektoch prudko stráca na výkonnosti. Pri scénach s viac ako 11 objektmi je recall takmer nulový, čo znamená, že SSD nedokáže spoľahlivo pracovať v zložitých podmienkach.

Na záver kombinovaná analýza (obrázok 5.12) ukazuje, že SSD najlepšie funguje pri jednoduchých scénach s veľkými objektmi. Avšak aj v týchto podmienkach dosahuje recall iba **0,28**, čo je výrazne menej ako pri YOLO alebo Faster R-CNN. Pri malých objektoch v preplnených scénach je recall takmer nulový, čím sa SSD stáva prakticky nepoužiteľným pre komplexnejšie úlohy detekcie objektov.



Obrázek 5.11: Presnosť podľa počtu objektov – SSD.



Obrázek 5.12: Kombinovaná analýza – SSD.

Zhrnutie analýzy presnosti

SSD je jednoznačne najslabším modelom zo všetkých troch. Má veľké problémy s detekciou malých objektov, nedokáže spoľahlivo identifikovať objekty v preplnených scénach a aj pri jednoduchších podmienkach zaostáva za YOLO a Faster R-CNN. Jeho najväčšou slabinou je veľmi nízky **recall** pri malých objektoch a výrazný pokles presnosti pri vyššom počte objektov na jednej scéne. Model sa preto javí ako nevhodný pre reálne aplikácie, kde sa predpokladá rôznorodosť objektov a komplexita scén.

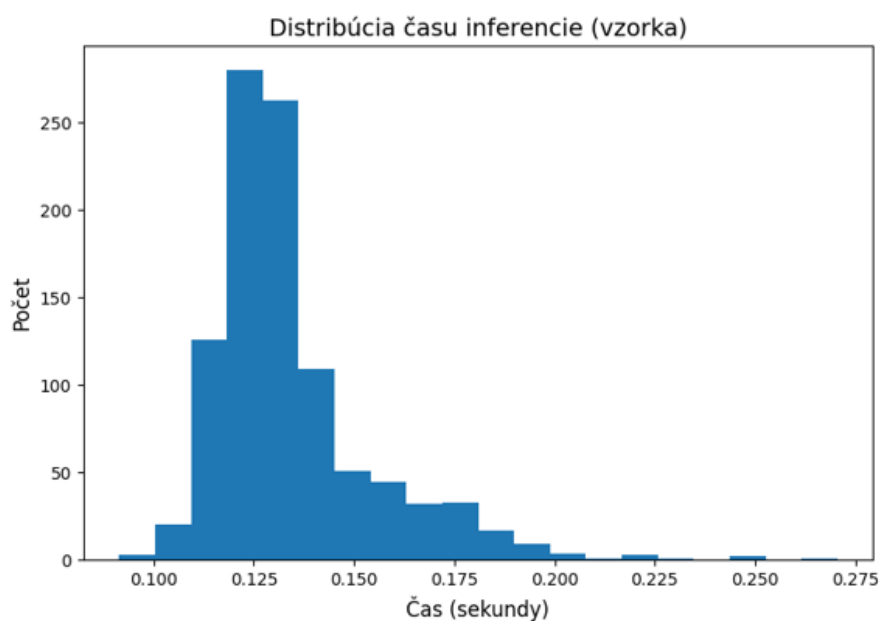
Naopak, **YOLO** (You Only Look Once) dosahuje vysoké hodnoty **precision** aj **recall** pri rôznych veľkostiach objektov. Je schopný pomerne spoľahlivo detegovať objekty aj v zložitejších podmienkach a jeho výkonnosť zostáva stabilná aj pri vyššom počte objektov na scéne. V porovnaní so SSD si udržiava výrazne vyššiu presnosť a robustnosť detekcie.

Faster R-CNN dosahuje najvyššiu presnosť zo všetkých testovaných modelov, predovšetkým pri malých objektoch, kde prekonáva aj YOLO. Jeho **recall** zostáva vysoký aj pri zložitejších scénach s viacerými objektmi. Vďaka presnému lokalizačnému mechanizmu dokáže spoľahlivo identifikovať objekty aj pri čiastočnom prekrytí alebo pri horších podmienkach v obraze.

5.2 Rýchlosť detekcie

Model YOLO

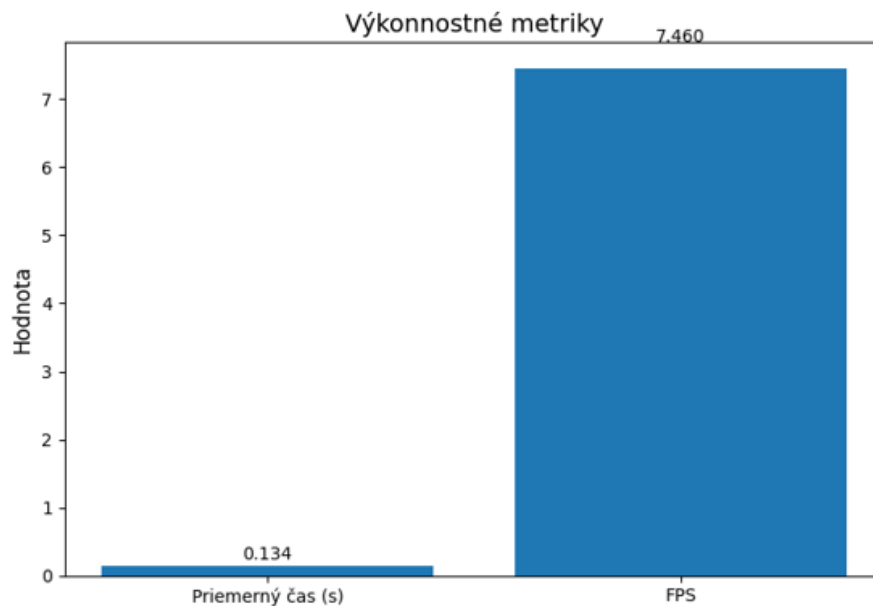
Dôležitou súčasťou analýzy detekčných modelov je aj hodnotenie výkonnostných metrík, ktoré odrážajú praktickú použiteľnosť modelu v reálnych aplikáciách. Výsledky výkonnosti modelu YOLO sú znázornené na obrázkoch 5.13 a 5.14.



Obrázek 5.13: Distribúcia času inferencie – YOLO (vzorka).

Na obrázku 5.13 je zobrazená distribúcia času inferencie pre jednotlivé obrázky vo vzorke testovacích dát. Väčšina inferencií sa pohybuje v intervale 0,12 – 0,14 sekundy, pričom maximum dosahuje histogram pri približne 0,125 sekundy. To naznačuje stabilnú a predvídateľnú výkonnosť modelu pri spracovaní obrázkov, bez výrazných výkyvov v čase výpočtu.

Obrázok 5.14 sumarizuje výkonnosť modelu pomocou dvoch hlavných metrík – **priemerného času inferencie** a **počtu spracovaných snímok za sekundu (FPS)**. Priemerný čas inferencie dosiahol hodnotu **0,134 sekundy**, čo je veľmi priaznivý výsledok pre použitie v reálnom čase.



Obrázek 5.14: Základné výkonnostné metriky – YOLO.

Zodpovedajúca hodnota FPS (frames per second) dosahuje **7,46 snímok za sekundu**, čo znamená, že model dokáže spracovať takmer 7 až 8 obrázkov každú sekundu.

Tieto hodnoty potvrdzujú, že model YOLO okrem slušnej detekčnej presnosti poskytuje aj spoľahlivý výkon z pohľadu rýchlosti.

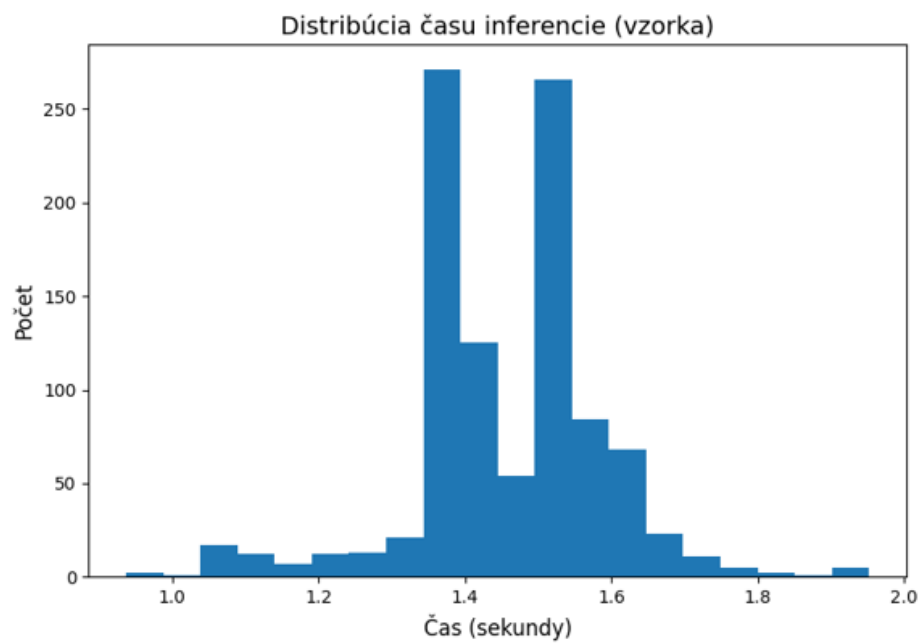
Model Faster R-CNN

Pri analýze modelu Faster R-CNN som použil rovnaké metriky ako pri modeli YOLO. Prvý graf (obrázok 5.15) znázorňuje histogram rozdelenia času spracovania jednej snímky – teda, ako dlho trvá modelu detekcia objektov na jednotlivých fotografiách.

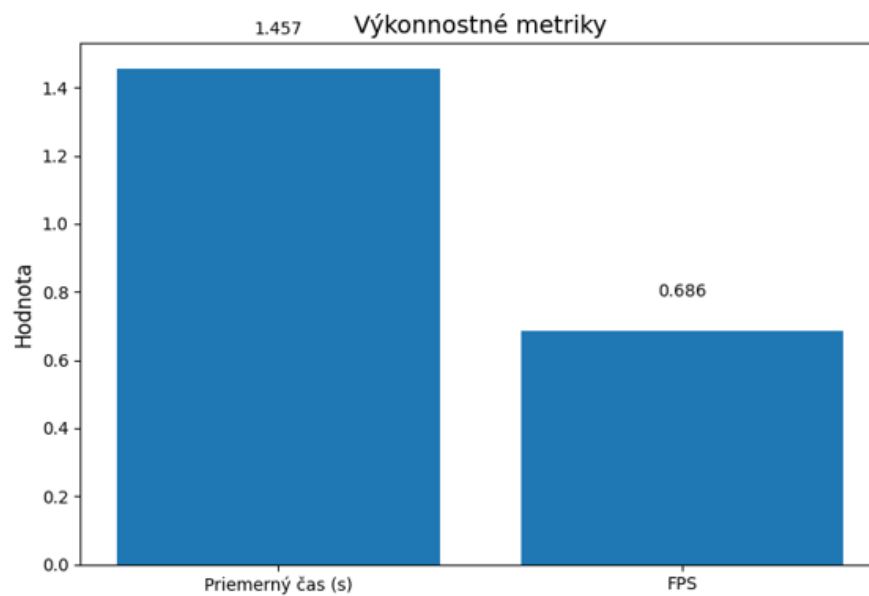
Z grafu vidieť, že hodnoty sú sústredené medzi 1,3 a 1,6 sekundy, s dvomi výraznými vrcholmi. Tento dvojvrcholový charakter distribúcie môže naznačovať, že model reaguje rôzne v závislosti od komplexnosti obrázkov – pri jednoduchších scénach má rýchlejšiu odozvu, pri zložitejších trvá inferencia dlhšie. V porovnaní s YOLO, ktorý dosahoval časy spracovania v rozmedzí 0,1 až 0,2 sekundy, je rozdiel výrazný a poukazuje na omnoho vyššiu výpočtovú náročnosť Faster R-CNN.

Graf na obrázku 5.16 obsahuje dve hodnoty: priemerný čas inferencie a počet spracovaných snímok za sekundu (FPS).

Priemerný čas spracovania jednej snímky dosiahol hodnotu **1,457 sekundy**, čo je v priamom kontraste s hodnotou **0,134 sekundy** pri YOLO. To sa odrazilo aj na počte snímok, ktoré dokáže model spracovať za sekundu — **iba 0,686 FPS**, v porovnaní so **7,46 FPS** pri YOLO. Model je teda výrazne pomalší ako model YOLO.



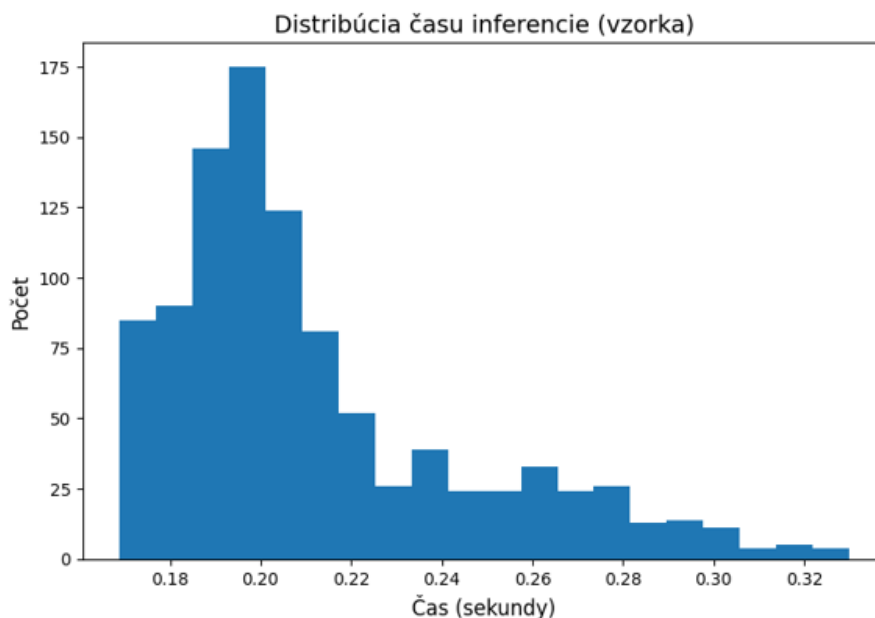
Obrázek 5.15: Distribúcia času inferencie – Faster R-CNN.



Obrázek 5.16: Výkonnostné metriky – Faster R-CNN.

Model SSD

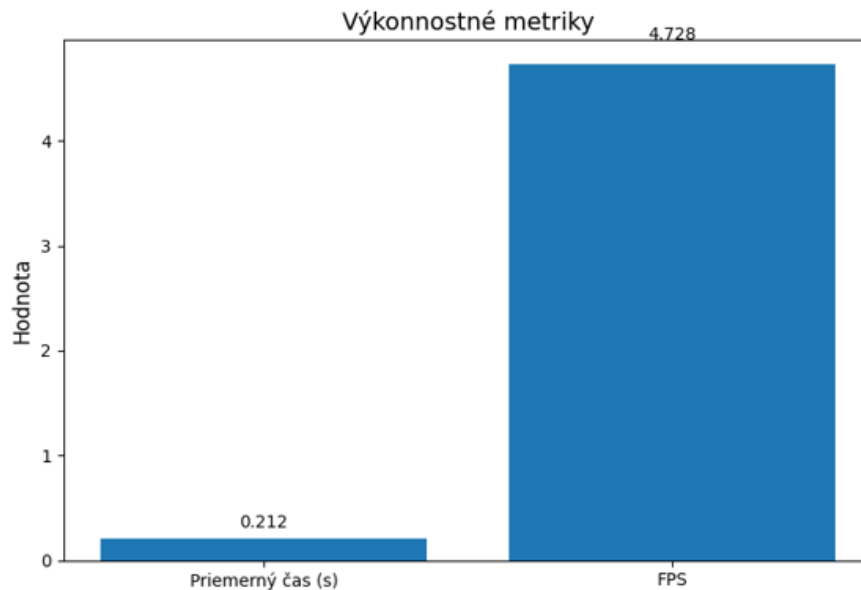
Model SSD (Single Shot Multibox Detector) som analyzoval rovnakým spôsobom ako predchádzajúce modely – cez distribúciu času inferencie a súhrnné výkonnostné metriky. Histogram na obrázku 5.17 znázorňuje, ako sa pohybuje doba spracovania jednej snímky počas testovania.



Obrázek 5.17: Distribúcia času inferencie – SSD.

Z grafu možno pozorovať, že väčšina inferencií trvala medzi 0,18 až 0,22 sekundy, pričom distribúcia má klesajúci charakter s miernym výskytom vyšších hodnôt do 0,32 sekundy. Na rozdiel od dvojvrcholovej distribúcie pri Faster R-CNN, SSD vykazuje hladší a stabilnejší priebeh, čo môže naznačovať konzistentnejšie správanie modelu nezávisle od zložitosti vstupných dát, to sa však popiera s predchádzajúcou analýzou presnosti tohoto modelu.

Z hľadiska výkonnostných metrík je SSD podobne ako YOLO rýchlejší ako Faster R-CNN. Priemerný čas spracovania jednej snímky je iba **0,212 sekundy** a počet spracovaných snímok za sekundu je **4,728 FPS**, ako je zobrazené na obrázku 5.18.



Obrázek 5.18: Výkonnostné metriky – SSD.

Zhrnutie analýzy rýchlosti detekcie

Z pohľadu rýchlosti inferencie vykazuje **YOLO** najlepšiu výkonnosť zo všetkých troch testovaných modelov. Priemerný čas spracovania jednej snímky dosiahol hodnotu **0,134 sekundy** a model dokáže spracovať približne **7,46 FPS** (obrázkov za sekundu). Táto hodnota je výrazne vyššia ako pri SSD aj Faster R-CNN, čím sa YOLO stáva najvhodnejším modelom pre aplikácie, kde je kľúčová rýchla odozva a real-time spracovanie.

SSD predstavuje kompromis medzi rýchlosťou a výpočtovou náročnosťou. Priemerný čas spracovania jednej snímky bol **0,212 sekundy**, a model dokáže spracovať **4,728 FPS**. Hoci nedosahuje úroveň rýchlosti modelu YOLO, stále ide o výrazne rýchlejšie riešenie než Faster R-CNN.

Faster R-CNN je z hľadiska rýchlosti najslabší model. Priemerný čas spracovania jednej snímky bol **1,457 sekundy**, pričom model dokázal spracovať iba **0,686 FPS**. Táto výrazne nižšia rýchlosť v porovnaní s YOLO a SSD je dôsledkom dvojfázového spracovania obrazu – najskôr generovania regionálnych návrhov a následnej klasifikácie. Hoci tento prístup vedie k vyššej presnosti detekcie, jeho výpočtová náročnosť je značne vyššia.

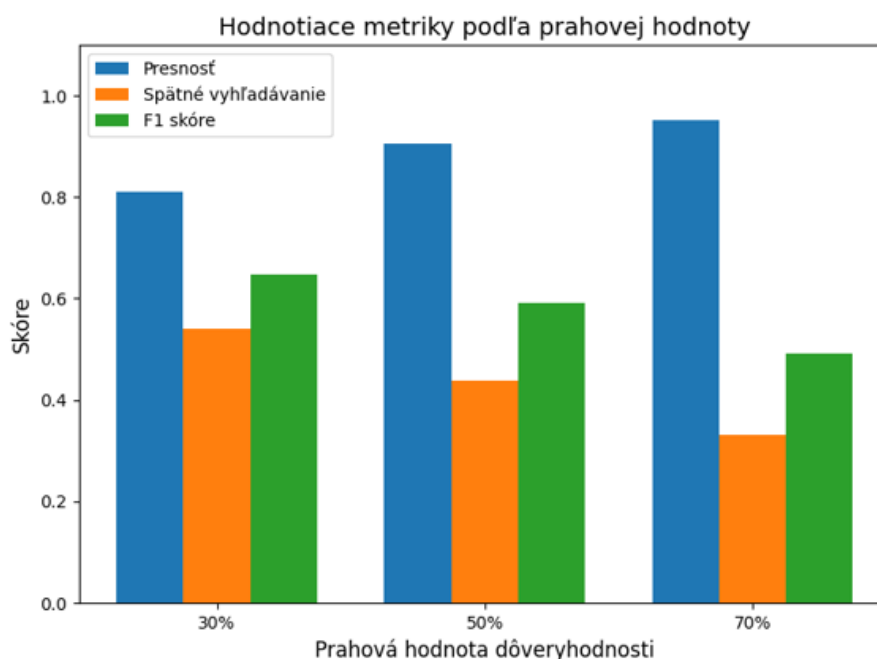
Tabulka 5.1: Porovnanie výkonnostných metrík pre jednotlivé modely.

Model	Priemerný čas (s)	FPS (snímok za sekundu)
YOLO	0,134	7,46
SSD	0,212	4,728
Faster R-CNN	1,457	0,686

5.3 Vplyv prahovej hodnoty dôveryhodnosti

V poslednej časti mojej analýzy som sa zamerlal na porovnávanie metrík pri rôznych prahových hodnotách dôveryhodnosti. Táto analýza poskytuje dôležité informácie o tom, ako sa mení presnosť, spätné vyhľadávanie a F1 skóre v závislosti od zvolenej prahovej hodnoty. Porovnal som aj počty pravých, falošných pozitív a negatív a nakoniec priemernú dôveryhodnosť v týchto prahových hodnotách. Výsledky tejto analýzy sú znázornené na nasledujúcich grafoch.

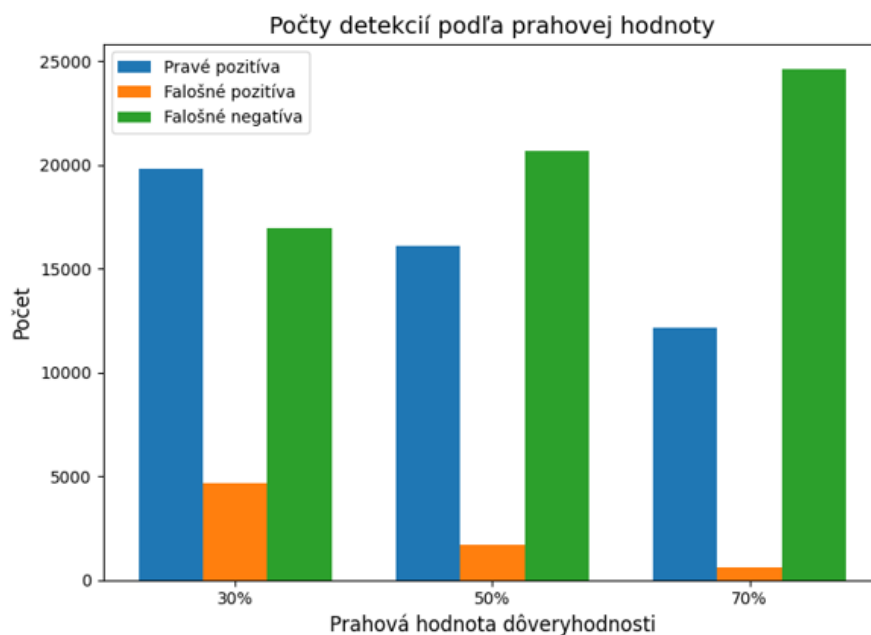
Model YOLO



Obrázek 5.19: Hodnotiace metriky podľa prahovej hodnoty dôveryhodnosti.

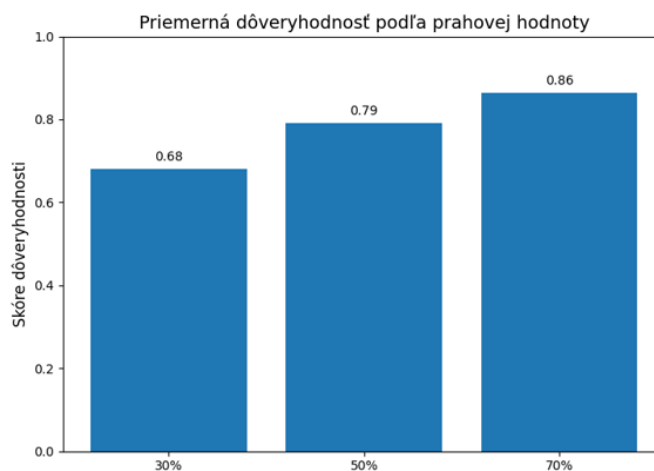
Na obrázku 5.19, je znázornené ako pri vyšších hodnotách prahu dochádza k nárastu **presnosti**, čo je spôsobené tým, že model YOLO sa stáva selektívnejším pri určovaní objektov. Tento prísnejší prístup však vedie k poklesu **spätného vyhľadávania**, pretože niektoré skutočné objekty už nespĺňajú zvýšený limit dôveryhodnosti a nie sú detegované. **F1 skóre**, ktoré vyjadruje rovnováhu medzi presnosťou a spätným vyhľadávaním, naznačuje, že optimálnu hodnotu môže predstavovať stredná prahová úroveň (50 %), kde sa dosahuje kompromis medzi elimináciou falošných detekcií a zachovaním dostatočného počtu správnych predikcií.

Ďalším dôležitým aspektom je sledovanie počtu pravých pozitívnych, falošne pozitívnych a falošne negatívnych detekcií v závislosti od prahovej hodnoty. Ako vidíme na obrázku 5.20, pri nízkej prahovej hodnote 30 % model vykazuje vysoký počet falošne pozitívnych detekcií, čo znamená, že niektoré objekty sú nesprávne označené ako prítomné. Naopak, pri zvýšení prahovej hodnoty na 70 % prudko rastie počet falošne negatívnych detekcií, pretože model začne ignorovať aj niektoré správne detegované objekty, ak ich dôveryhodnosť



Obrázek 5.20: Počty detekcií podľa prahovej hodnoty dôveryhodnosti.

nie je dostatočne vysoká. Optimálnym riešením je nájsť rovnováhu medzi týmito dvoma extrémami, čo v tomto prípade naznačuje práve prahová hodnota 50 %.



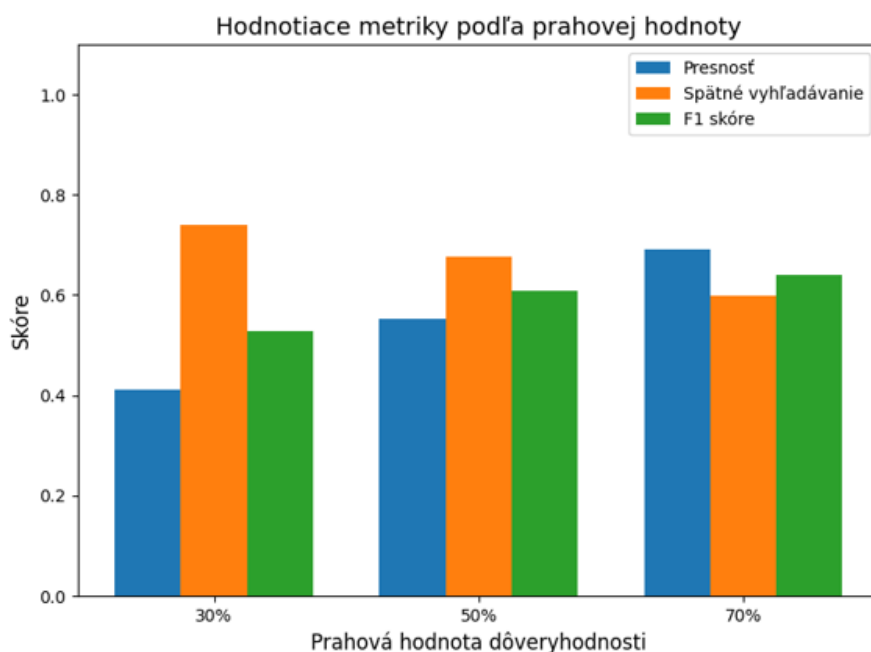
Obrázek 5.21: Priemerná dôveryhodnosť podľa prahovej hodnoty.

Model YOLO priraduje každej detekcii skóre dôveryhodnosti, ktoré odráža mieru presvedčenia modelu o správnosti predikcie. Ako znázorňuje obrázok 5.21, pri nízkej prahovej hodnote 30 % dosahuje priemerné skóre dôveryhodnosti hodnotu približne **0,68**, zatiaľ čo pri zvýšení prahu na 50 % sa zvyšuje na **0,79**. Pri najvyššej testovanej prahovej hodnote 70 % je priemerné skóre dôveryhodnosti ešte vyššie, dosahuje úroveň **0,86**. Tento trend je očakávaný, keďže prísnejší filter ponecháva iba predikcie, pri ktorých má model vyššiu mieru istoty.

Na základe vykonanej analýzy je zrejmé, že výber prahovej hodnoty má zásadný vplyv na správanie modelu YOLO. Nízke hodnoty vedú k vyššiemu počtu chybných detekcií, zatiaľ čo príliš vysoké prahy spôsobujú stratu dôležitých informácií. Optimálna prahová hodnota sa preto nachádza niekde medzi týmito extrémami, kde model dosahuje dobrý kompromis medzi presnosťou a spätným vyhľadávaním.

Model Faster R-CNN

Podobne ako pri modeli YOLO, aj tento model pracuje s pravdepodobnostným vyhodnotením detegovaných objektov, pričom výber vhodnej prahovej hodnoty zásadne ovplyvňuje kvalitu výsledkov. V mojej analýze skúmam opäť kľúčové metriky - **presnosť**, **spätne vyhľadávanie** a **F1 skóre** pri troch prahových hodnotách: 30%, 50% a 70%.

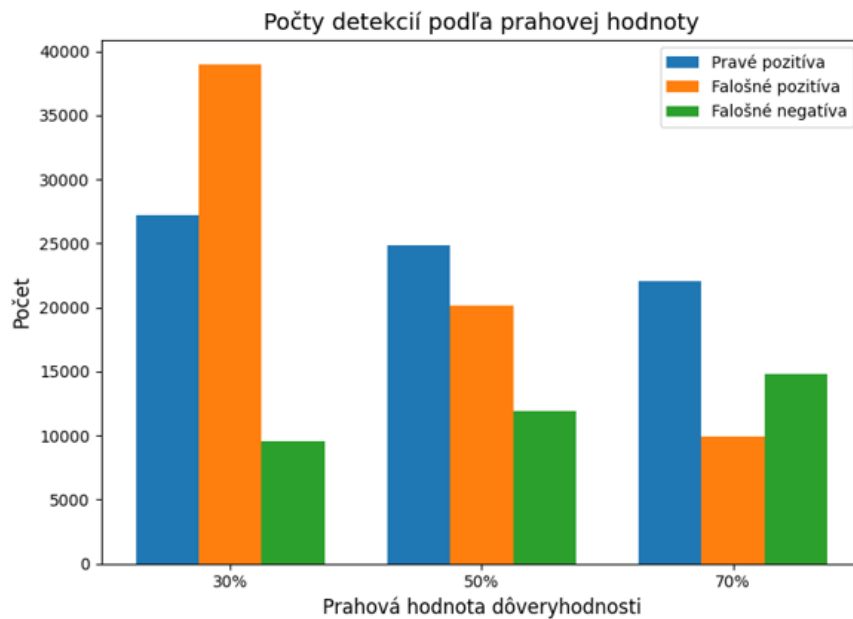


Obrázek 5.22: Hodnotiace metriky podľa prahovej hodnoty dôveryhodnosti.

Na obrázku 5.22, je možné vidieť, že model Faster R-CNN vykazuje podobný trend ako YOLO - so zvyšujúcou sa prahovou hodnotou rastie presnosť detekcie, zatiaľ čo spätne vyhľadávanie klesá. Pri prahovej hodnote 30% dosahuje model relatívne nízku presnosť (približne 0,41), ale vysoké spätne vyhľadávanie (približne 0,74). S rastúcou prahovou hodnotou sa presnosť výrazne zlepšuje, pričom pri 70% dosahuje hodnotu približne 0,69.

Na rozdiel od modelu YOLO, ktorý vykazoval najlepšie F1 skóre pri nízkej prahovej hodnote 30%, Faster R-CNN dosahuje mierne vyššie F1 skóre pri prahovej hodnote 70%. Tento rozdiel naznačuje, že Faster R-CNN dokáže lepšie využiť vyššie prahové hodnoty bez výraznej straty skutočných detekcií, čo pripisujem dvojfázovému prístupu tohto modelu pri detekcii objektov.

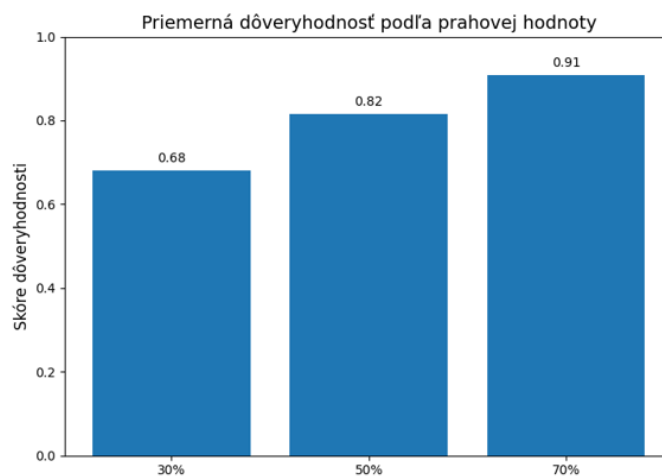
Z obrázku 5.23 je zrejmé, že pri nízkej prahovej hodnote 30% model produkuje významné množstvo falošne pozitívnych detekcií. Zaujímavým pozorovaním je, že pri strednej prahovej hodnote 50% dochádza k výraznému poklesu falošne pozitívnych detekcií, zatiaľ čo počet



Obrázek 5.23: Počty detekcií podľa prahovej hodnoty dôveryhodnosti.

pravých pozitívnych detekcií klesá miernejšie. To naznačuje, že model pri nízkej prahovej hodnote deteguje podstatne väčší počet objektov, ktoré na obrázku v porovnaní s "Ground truth" (objekty manuálne nájdené na obrázku, ako súčasť datasetu), pravdepodobne nie sú. Model YOLO to mal v tomto prípade presne naopak a vyhýbal sa detegovaniu falošných pozitív.

Pri už spomínanom porovnaní s modelom YOLO si všímam, že Faster R-CNN vykazuje mierne lepšiu schopnosť udržať si pravé pozitívne detekcie aj pri vyšších prahových hodnotách. Toto sa dá vysvetliť sofistikovanejším mechanizmom detekcie, ktorý zahŕňa najprv generovanie návrhov regiónov (region proposals) a následne klasifikáciu objektov v týchto regiónoch. Na druhej strane, YOLO, ktorý je známy svojou rýchlosťou, používa jednoduchší prístup, čo môže viesť k väčšej strate detekcií pri vyšších prahových hodnotách.



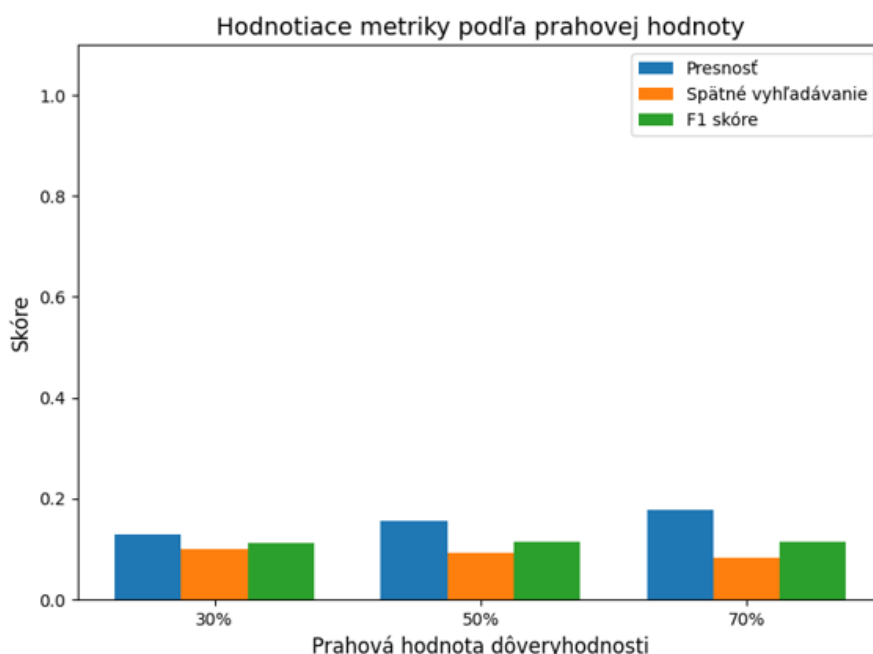
Obrázek 5.24: Priemerná dôveryhodnosť podľa prahovej hodnoty.

Poslednou metrikou je podobne ako pri modeli YOLO, zmena priemernej dôveryhodnosti detekcií so zvyšovaním prahovej hodnoty. Ako znázorňuje obrázok 5.24, pri prahovej hodnote 30% je priemerná dôveryhodnosť na úrovni 0,68, pri 50% stúpa na 0,82 a pri 70% dosahuje až 0,91.

V porovnaní s modelom YOLO, Faster R-CNN vykazuje podobné, no trochu lepšie hodnoty priemernej dôveryhodnosti, čomu určite napomáha komplexnejší mechanizmus detekcie, ktorý model Faster R-CNN využíva.

Model SSD

V tejto časti sa zameriam na analýzu modelu SSD (Single Shot Detector) a jeho správanie pri rôznych prahových hodnotách dôveryhodnosti. SSD predstavuje jednoetapový detektor podobne ako YOLO, ale podľa mojich pozorovaní vykazuje výrazne odlišné charakteristiky výkonnosti.

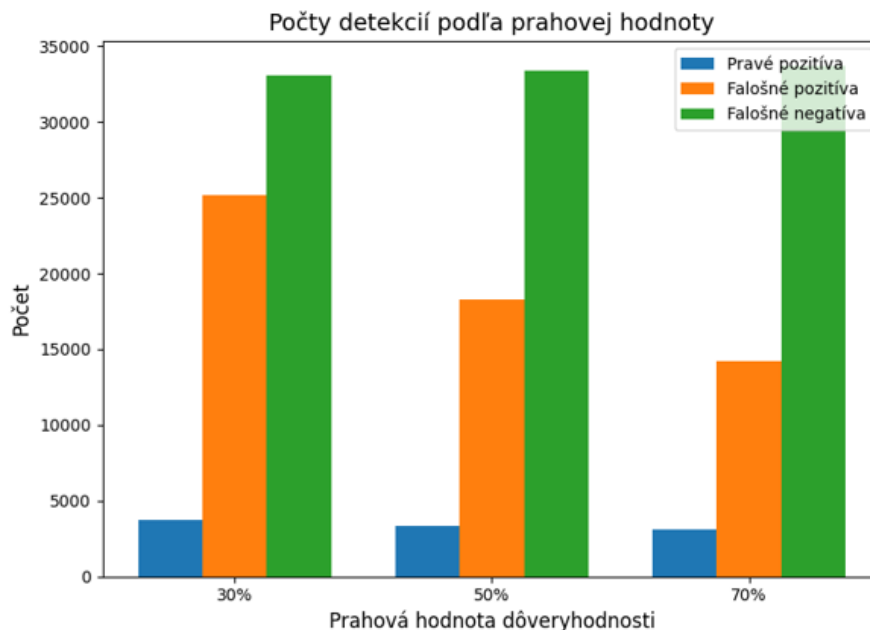


Obrázok 5.25: Hodnotiace metriky podľa prahovej hodnoty dôveryhodnosti.

Z grafov je zrejmé, že model SSD dosahuje výrazne nižšie hodnoty všetkých kľúčových metrík v porovnaní s modelmi YOLO a Faster R-CNN. Presnosť modelu dosahuje pri 30% prahovej hodnote iba približne 0,17, čo je podstatne menej ako pri ostatných modeloch. So zvyšovaním prahovej hodnoty síce presnosť rastie, ale aj pri 70% zostáva na pomerne nízkej úrovni približne 0,19.

Spätné vyhľadávanie vykazuje ešte výraznejší prepád, pričom už pri 30% prahovej hodnote dosahuje iba 0,15 a pri 70% klesá až na 0,13. Táto kombinácia nízkej presnosti a nízkeho spätného vyhľadávania vedie k veľmi nízkemu F1 skóre v celom rozsahu prahových hodnôt, čo naznačuje celkovo slabú výkonnosť modelu.

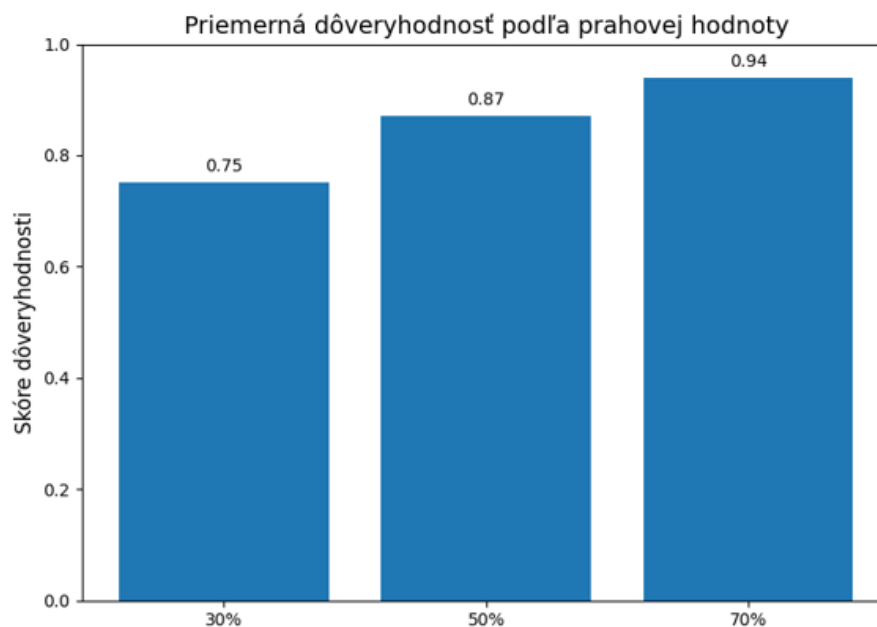
Pri pohľade na počty detekcií je jasne viditeľný hlavný problém modelu SSD - extrémne nízky počet pravých pozitívnych detekcií. Už pri najnižšej prahovej hodnote 30% model



Obrázek 5.26: Počty detekcií podľa prahovej hodnoty dôveryhodnosti.

identifikuje iba približne 4000 pravých pozitív, čo je výrazne menej ako 27000 pri YOLO alebo Faster R-CNN. S rastúcou prahovou hodnotou tento počet ďalej klesá.

Zároveň si všimam, že model SSD produkuje pomerne vysoký počet falošne pozitívnych detekcií pri nízkej prahovej hodnote. Ďalším problémom je veľký počet negatívnych detekcií, ktorý naznačuje, že model nedokáže spoľahlivo identifikovať skutočné objekty na snímkach.



Obrázek 5.27: Priemerná dôveryhodnosť podľa prahovej hodnoty.

Zaujímavým pozorovaním je, že aj napriek celkovo slabej výkonnosti model SSD vykazuje podobný trend rastu priemernej dôveryhodnosti s rastúcou prahovou hodnotou ako predchádzajúce modely. Pri prahovej hodnote 30% dosahuje priemernú dôveryhodnosť približne **0,75**, pri 50% približne **0,87** a pri 70% približne **0,94**.

Tieto hodnoty sú porovnateľné modeloch YOLO a Faster R-CNN, to však môže byť dôsledkom celkovo nízkeho počtu detekcií. To naznačuje, že model SSD si "dôveruje" takmer rovnako ako ostatné modely, hoci jeho skutočná výkonnosť je podstatne nižšia. Tento jav môže poukazovať na nedostatočnú kalibráciu odhadov dôveryhodnosti modelu SSD.

Zhrnutie analýzy vplyvu prahovej hodnoty na detekciu

Na základe mojej analýzy môžem porovnať všetky tri skúmané modely - YOLO, Faster R-CNN a SSD:

- **Presnosť detekcie:** Fastest R-CNN dosahuje najvyššiu presnosť, najmä pri vyšších prahových hodnotách. YOLO vykazuje dobrú presnosť pri stredných a vyšších prahových hodnotách, zatiaľ čo SSD zaostáva v presnosti pri všetkých prahových hodnotách.
- **Spätné vyhľadávanie:** YOLO dosahuje najlepšie výsledky pri nižších prahových hodnotách, Faster R-CNN si udržiava relatívne stabilné spätné vyhľadávanie naprieč rôznymi prahovými hodnotami, zatiaľ čo SSD vykazuje výrazne nízke hodnoty spätného vyhľadávania pri všetkých prahových hodnotách.
- **Falošné detekcie:** YOLO produkuje najmenej falošne pozitívnych detekcií, Faster R-CNN generuje extrémne vysoký počet falošne pozitívnych detekcií pri nízkych prahových hodnotách, ale tento počet prudko klesá s rastúcou prahovou hodnotou. SSD produkuje relatívne vysoký počet falošne pozitívnych detekcií, ale najviac problematický je jeho vysoký počet falošne negatívnych detekcií.
- **Priemerná dôveryhodnosť:** Všetky tri modely vykazujú podobný trend rastu priemernej dôveryhodnosti s rastúcou prahovou hodnotou, pričom Faster R-CNN a YOLO dosahujú mierne vyššie hodnoty než SSD.

Moja analýza ukazuje, že výber vhodnej prahovej hodnoty dôveryhodnosti má zásadný vplyv na výkonnosť všetkých troch skúmaných modelov - YOLO, Faster R-CNN a SSD. Pri nižších prahových hodnotách modely detegujú viac objektov, ale zároveň produkujú viac falošných detekcií. Naopak, pri vyšších prahových hodnotách sa zvyšuje presnosť, ale za cenu ignorovania niektorých skutočných objektov.

Na základe analyzovaných metrík môžem odporučiť, že pre model YOLO je vhodnejšie použiť strednú prahovú hodnotu (50%), zatiaľ čo Faster R-CNN môže benefitovať z vyššej prahovej hodnoty (70%). Pre model SSD je ťažké nájsť optimálnu prahovú hodnotu vzhľadom na jeho celkovo slabú výkonnosť, ale ak by sa mal použiť, pravdepodobne by bola najvhodnejšia nižšia prahová hodnota (30%), ktorá aspoň čiastočne kompenzuje jeho nízke spätné vyhľadávanie.

Z porovnaní všetkých troch modelov jednoznačne vyplýva, že Faster R-CNN predstavuje najspoľahlivejšiu voľbu pre presné detekcie, YOLO ponúka dobrý kompromis medzi presnosťou a rýchlosťou, zatiaľ čo SSD výrazne zaostáva v kľúčových metrikách a jeho použitie by som odporučil iba v prípadoch, kde sú kritické extrémne nízke výpočtové nároky.

5.4 Záverečné porovnanie modelov

V rámci tejto práce som analyzoval tri detekčné modely – YOLO, Faster R-CNN a SSD – z pohľadu rôznych aspektov ich výkonnosti a presnosti. Na základe komplexnej analýzy všetkých metrík môžem sformulovať nasledujúce závery o každom z modelov.

Model **Faster R-CNN** sa prejavil ako najspoľahlivejší z hľadiska presnosti detekcie. Jeho dvojfázový prístup zahŕňajúci najprv generovanie návrhov regiónov a následne ich klasifikáciu mu umožňuje dosahovať vynikajúce výsledky v širokom spektre scenárov. Najvýraznejšou prednosťou tohto modelu je schopnosť spoľahlivo detegovať malé objekty – v kategóriách **Tiny** a **Small** detegoval výrazne viac objektov než konkurenčné modely. Pri komplexných scénach s viacerými objektmi si udržiava stabilnú výkonnosť a dosahuje vyrovnané hodnoty presnosti aj spätného vyhľadávania. Pri vyšších prahových hodnotách dôveryhodnosti (70%) dosahuje najvyššie F1 skóre, čo poukazuje na jeho schopnosť generovať kvalitné a spoľahlivé predikcie. Jeho najvýraznejšou nevýhodou je však rýchlosť – s priemerným časom inferencie 1,457 sekundy a výkonom len 0,686 FPS je približne desaťkrát pomalší než YOLO, čo výrazne limituje jeho použiteľnosť v aplikáciách vyžadujúcich spracovanie v reálnom čase.

Model **YOLO** predstavuje vyvážený kompromis medzi presnosťou a rýchlosťou. Dosahuje slušnú presnosť detekcie pri väčších objektoch, no zaostáva za Faster R-CNN pri detekcii malých objektov (kategórie **Tiny** a **Small**). Pri preplnených scénach s viacerými objektmi postupne klesá jeho výkonnosť, najmä z pohľadu spätného vyhľadávania. Najsilnejšou stránkou modelu YOLO je jeho rýchlosť – s priemerným časom inferencie len 0,134 sekundy a výkonom 7,46 FPS prekonáva ostatné testované modely a predstavuje vhodnú voľbu pre aplikácie vyžadujúce rýchlu odozvu. Z hľadiska prahovej hodnoty dôveryhodnosti dosahuje YOLO optimálnu výkonnosť pri strednej hodnote (50%), kde nachádza najlepšiu rovnováhu medzi presnosťou a spätným vyhľadávaním. Zaujímavou charakteristikou modelu YOLO je relatívne nízky počet falošne pozitívnych detekcií, čo znamená, že ak model niečo deteguje, s vysokou pravdepodobnosťou ide o skutočný objekt.

Model **SSD** sa v mojej analýze prejavil ako najslabší zo všetkých troch porovnávaných modelov. Dosahuje výrazne nižšie hodnoty presnosti, spätného vyhľadávania aj F1 skóre pri všetkých testovaných scenároch. Najvýraznejšie zaostáva pri detekcii malých objektov, kde je jeho výkonnosť extrémne nízka. Rovnako pri komplexných scénach s viacerými objektmi jeho výkonnosť prudko klesá. Z hľadiska rýchlosti predstavuje SSD strednú cestu – s priemerným časom inferencie 0,212 sekundy a výkonom 4,728 FPS je pomalší než YOLO, ale výrazne rýchlejší než Faster R-CNN. Problematickou charakteristikou modelu SSD je vysoký počet falošne negatívnych detekcií a celkovo nízky počet pravých pozitívnych detekcií, čo znamená, že model nedokáže spoľahlivo identifikovať objekty prítomné na obraze. Paradoxne, aj napriek slabej výkonnosti vykazuje SSD vysoké hodnoty priemernej dôveryhodnosti, čo môže naznačovať nedostatočnú kalibráciu modelu.

Kapitola 6

Záver

V závere mojej bakalárskej práce by som rád zhrnul celý proces vývoja aplikácie pre správu fotiek a editáciu metadát. Primárnym cieľom bolo vytvoriť zabezpečenú webovú aplikáciu založenú na frameworku Django, ktorá by užívateľom umožnila efektívne spravovať svoje fotografie a zároveň by využívala moderné metódy detekcie objektov na automatické značkovanie obsahu fotografií. Táto funkcionality značne zjednodušuje proces organizácie a vyhľadávania fotografií podľa ich obsahu, čím prináša praktickú hodnotu pre bežných užívateľov.

Dôležitou súčasťou práce bola analýza a implementácia troch najpopulárnejších modelov na detekciu objektov: YOLO (You Only Look Once), Faster R-CNN (Region-based Convolutional Neural Network) a SSD (Single Shot Detector) do webovej aplikácie. Podrobná analýza týchto modelov, ktorej som venoval predchádzajúcu kapitolu, mi poskytla cenné poznatky, ktoré som následne využil pri integrácii týchto modelov do praktického riešenia.

Implementácia modelov do webovej aplikácie založenej na frameworku Django tvorila praktickú časť mojej práce. Vytvoril som zabezpečený systém, kde si užívatelia môžu organizovať svoje fotografie do galérií, pričom pri nahrávaní fotografií systém automaticky deteguje objekty na obraze a priradí im príslušné značky (tagy) s použitím jedného (prípadne viacerých súčasne) modelov na detekciu. Tieto tagy sú následne uložené do metadát fotografií a môžu byť využité na rýchle vyhľadávanie a filtrovanie obsahu galérie. Užívateľské rozhranie aplikácie poskytuje intuitívnu vizualizáciu detegovaných objektov pomocou ohraničujúcich rámciekov (bounding boxes), pričom užívatelia majú možnosť tieto detekcie upravovať, odstraňovať alebo pridávať nové. Implementoval som tiež funkcionality, ktorá umožňuje stiahnuť fotografiu s upravenými metadátami obsahujúcimi všetky priradené tagy.

Implementácia Django frameworku ako backendu aplikácie zabezpečila robustné spracovanie užívateľských požiadaviek, bezpečnú správu účtov a efektívnu integráciu s neuronovými sieťami. Využitie moderných knižníc ako PyTorch, Torchvision, Ultralytics a OpenCV umožnilo plynulú implementáciu a optimalizáciu detekčných algoritmov.

Ako fotograf verím, že moja bakalárska práca prináša praktický nástroj, ktorý môže užívateľom výrazne zjednodušiť organizáciu a vyhľadávanie vo fotografických galériách. Kombinácia moderných technológií detekcie objektov s užívateľsky prívetivým rozhraním vytvára aplikáciu, ktorá rozširuje možnosti správy digitálnych fotografií a zároveň udržiava plnú kontrolu užívateľa nad svojimi dátami.

Do budúcnosti plánujem rozšíriť aplikáciu o viaceré pokročilé funkcionality. Jednou z nich je implementácia rozpoznávania tvárí, ktoré by umožnilo automatické označovanie osôb na fotografiách a ich následné filtrovanie. Ďalším plánovaným vylepšením je integrácia systému pre klasifikáciu celých scén, ktorý by dokázal automaticky kategorizovať fotografie podľa

typu prostredia (mesto, príroda, interiér) alebo aktivity (šport, oslava, dovolenka). Zaujímavým rozšírením by mohla byť aj implementácia pokročilej analýzy metadát, ktorá by dokázala automaticky vytvárať albumy na základe času, miesta alebo udalostí.

Literatura

- [1] ADOBE. *Adobe Lightroom* online. 2024. Dostupné z: <https://lightroom.adobe.com/>. [cit. 2024-11-30].
- [2] ADOBE. *What is an EXIF file?* online. 2024. Dostupné z: <https://www.adobe.com/cz/creativecloud/file-types/image/raster/exif-file.html>. [cit. 2024-11-30].
- [3] ALPAYDIN, E. *Introduction to Machine Learning*. London: The MIT Press, 2010. ISBN 978-0-262-01243-0.
- [4] AUTONOMOUS, T. *Anchor Boxes in Object Detection* online. 2024. Dostupné z: <https://www.thinkautonomous.ai/blog/anchor-boxes/>. [cit. 2024-11-30].
- [5] AZEEM, I. *Understanding ResNet Architecture: A Deep Dive into Residual Neural Network* online. 2024. Dostupné z: <https://medium.com/@ibtedaazeem/understanding-resnet-architecture-a-deep-dive-into-residual-neural-network-2c792e6537a9>. [cit. 2024-10-22].
- [6] BROWNLEE, J. *Supervised and Unsupervised Machine Learning Algorithms* online. 2024. Dostupné z: <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>. [cit. 2024-10-31].
- [7] DARKTABLE. *Darktable: Open Source Photography Workflow Software* online. 2024. Dostupné z: <https://www.darktable.org/>. [cit. 2024-11-30].
- [8] ENGLISH, P. *Object Detection Using the Faster R-CNN Model with ResNet-50 Architecture in PyTorch* online. 2024. Dostupné z: <https://python.plainenglish.io/object-detection-using-the-faster-r-cnn-model-with-resnet-50-architecture-in-pytorch-7019a37034c7>. [cit. 2024-11-30].
- [9] EUROPE, C. *All About EXIF* online. 2024. Dostupné z: https://www.canon-europe.com/pro/infobank/all-about-exif/?utm_source=chatgpt.com. [cit. 2024-11-30].
- [10] FORMAT. *Photo Metadata: What It Is and How to Use It* online. 2024. Dostupné z: https://www.format.com/magazine/photo-metadata?utm_source=chatgpt.com. [cit. 2024-11-30].
- [11] GIRSHICK, R. B. Fast R-CNN. *ArXiv*, 2015. Dostupné z: <https://arxiv.org/pdf/1504.08083>.
- [12] HARVEY, P. *ExifTool by Phil Harvey* online. 2024. Dostupné z: <https://exiftool.org/>. [cit. 2024-11-30].

- [13] HE, K.; ZHANG, X.; REN, S. a SUN, J. Deep Residual Learning for Image Recognition. *ArXiv*, 2015. Dostupné z: <https://arxiv.org/pdf/1506.02640>.
- [14] INDOML. *Student Notes: Convolutional Neural Networks (CNN) Introduction* online. 2018. Dostupné z: <https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/>. [cit. 2024-11-03].
- [15] IPTC. *Photo Metadata* online. 2024. Dostupné z: https://iptc.org/standards/photo-metadata/photo-metadata/?utm_source=chatgpt.com. [cit. 2024-11-30].
- [16] JESSE. *Single Shot MultiBox Detector (SSD) Explained by a J.D.* online. 2024. Dostupné z: <https://medium.com/@jesse419419/single-shot-multibox-detector-ssd-explained-by-a-j-dda10ba42a29>. [cit. 2024-11-30].
- [17] JÚLIA KUČEROVÁ, E. Šikudová a Zuzana Cerneková a Vanda Benešová a Zuzana Haladová a. *Počítačové videnie: Detekcia a rozpoznávanie objektov*. 1. vyd. Livornská 445, 109 00 Praha 10: Wikina, 2011. 397 s. ISBN 978-80-87925-07-2.
- [18] LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S. et al. SSD: Single Shot MultiBox Detector. *ArXiv*, 2016. Dostupné z: <https://arxiv.org/abs/1512.02325>.
- [19] MCFEE, B. *Linear Shift-Invariant Systems* online. 2024. Dostupné z: <https://brianmcfree.net/dstbook-site/content/ch03-convolution/LSI.html>. [cit. 2024-11-30].
- [20] PYIMAGESEARCH. *Intersection over Union (IoU) for Object Detection* online. 2016. Dostupné z: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/?utm_source=chatgpt.com. [cit. 2024-11-30].
- [21] REN, S.; HE, K.; GIRSHICK, R. B. a SUN, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *ArXiv*, 2015. Dostupné z: <https://arxiv.org/pdf/1506.01497>.
- [22] REVUE, K. K. *Bibliografická metadata v sémantickém webu* online. 2019. Dostupné z: https://knihovnarevue.nkp.cz/archiv/2019-2/recenzovane-prispevky/bibliograficka-metadata-v-semantickem-webu?utm_source=chatgpt.com. [cit. 2024-11-30].
- [23] ROBOFLOW. *Guide to YOLO Models* online. 2024. Dostupné z: https://blog.roboflow.com/guide-to-yolo-models/?utm_source=chatgpt.com. [cit. 2024-11-30].
- [24] SCIENCE, T. D. *Types of Machine Learning Algorithms You Should Know* online. 2024. Dostupné z: <https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861>. [cit. 2024-10-18]. Accessed: 2024-11-27.
- [25] SERGEY IOFFE, C. S. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ArXiv preprint arXiv:1502.03167*, 2015. Dostupné z: <https://arxiv.org/abs/1502.03167>.
- [26] SIMONYAN, K. a ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv*, 2014. Dostupné z: <https://arxiv.org/pdf/1405.0312>.

- [27] TSUNG YI LIN, R. B. G. K. H. P. D. Focal Loss for Dense Object Detection. *ArXiv preprint arXiv:1708.02002*, 2017. Dostupné z: <https://arxiv.org/abs/1708.02002>.
- [28] ULTRALYTICS. *Ultralytics Documentation* online. 2024. Dostupné z: <https://docs.ultralytics.com/>. [cit. 2024-11-08].
- [29] VIDHYA, A. *Build ResNet from Scratch with Python* online. 2021. Dostupné z: <https://www.analyticsvidhya.com/blog/2021/06/build-resnet-from-scratch-with-python/>. [cit. 2024-11-30].
- [30] WU, J. *Introduction to Convolutional Neural Networks* online. 2017. Dostupné z: <https://cs.nju.edu.cn/wujx/paper/CNN.pdf>. [cit. 2024-11-30].