

UNIVERZITA PARDUBICE  
FAKULTA ELEKTROTECHNIKY  
A INFORMATIKY

DIPLOMOVÁ PRÁCE

2024

Bc. Anežka Blažková

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

**AUTOMATICKÁ DETEKCE ODPADU V OBRAZOVÝCH  
DATECH**

Bc. Anežka Blažková

Diplomová práce 2024

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2022/2023

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Anežka Blažková**  
Osobní číslo: **I21301**  
Studijní program: **N0714A150005 Automatické řízení**  
Téma práce: **Automatická detekce odpadu v obrazových datech**  
Zadávací katedra: **Katedra řízení procesů**

## Zásady pro vypracování

Cílem práce je návrh a implementace systému pro automatickou detekci volně ležícího odpadu. Systém bude využívat obrazová data sledovaného prostoru, ve kterých bude detekovat objekty vhodné k odebrání. Na základě informace o relativní pozici objektu vůči kamerovému snímači bude systém poskytovat informace k řízení pohybu mobilního robota pro zajištění možného odběru odpadu. Zpracování a vyhodnocení obrazových dat bude založeno na technologii konvolučních sítí. V rámci práce bude vytvořena aplikace pro vyhodnocování obrazových dat dále poskytující plán pohybu přiblížení k odpadu určenému k odběru.

**Teoretická část:** Stručná rešerše existujících nástrojů pro detekci a lokalizaci objektů v obrazových datech založených na metodách umělé inteligence. Popis senzorové techniky pro sběr dat. Metodika sběru dat. Popis softwarových nástrojů použitých pro řešení praktické části.

**Praktická část:** Návrh a implementace systému pro sběr a vyhodnocování dat. Sběr a analýza dat. Návrh a implementace softwarového nástroje založeného na vybraném paradigmatu umělých neuronových sítí. Testování a zhodnocení systému pomocí běžných metrik používaných pro hodnocení neuronových sítí. Dokumentace softwaru včetně testovacího scénáře demonstrujícího použití.

Rozsah pracovní zprávy: **cca 50 stran**  
Rozsah grafických prací:  
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

GOODFELLOW, Ian, Yoshua BENGIO a Aaron COURVILLE. *Deep learning*. Cambridge, Massachusetts: The MIT Press, [2016]. ISBN 978-02-620-3561-3.  
GONZALEZ, Rafael C. a Richard E. WOODS. *Digital image processing*. Fourth edition. New York: Pearson, [2018]. ISBN 978-013-3356-724.

Vedoucí diplomové práce: **Ing. Dominik Štursa**  
Katedra řízení procesů

Datum zadání diplomové práce: **8. listopadu 2022**  
Termín odevzdání diplomové práce: **19. května 2023**

L.S.

---

**Ing. Zdeněk Němec, Ph.D.**  
děkan

---

**Ing. Daniel Honc, Ph.D.**  
vedoucí katedry

V Pardubicích dne 15. listopadu 2022

## **Prohlášení autora**

Prohlašuji:

Tuto práci jsem vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 15. 8. 2024

Bc. Anežka Blažková

### **Poděkování**

Ráda bych poděkovala Ing. Dominikovi Štursovi za vedení diplomové práce, za cenné rady, konzultace a trpělivost. Dále bych ráda poděkovala rodině a přátelům, kteří mě podporovali po celý průběh studia.

V Pardubicích dne 15. 8. 2024

Bc. Anežka Blažková

## **ANOTACE**

*Cílem diplomové práce je vytvořit software pro automatickou detekci odpadu v prostředí Python pomocí konvolučních neuronových sítí. Teoretická část je věnována problematice odpadu a recyklace. Práce popisuje strukturu konvoluční neuronové sítě a její učení. Dále se zabývá algoritmy a metriky pro klasifikaci a detekci odpadu. Praktická část obsahuje dataset se shromážděnými daty a jejich augmentace, které jsou využity k následnému trénování a optimalizaci modelu. V práci je zvolen algoritmus YOLOv5 a jeho modifikace YOLOv5n, YOLOv5s a YOLOv5m, na kterých je síť postupně trénována a následně dle zvolených metrik vyhodnocena. Součástí je také jednoduchá aplikace, pomocí které lze jednoduše vyhodnotit snímky.*

## **KLÍČOVÁ SLOVA**

*Odpad, Python, Konvoluční neuronové sítě, Zpracování obrazu, YOLOv5*

## **TITLE**

*AUTOMATIC WASTE DETECTION IN IMAGE DATA*

## **ANNOTATION**

*The goal of this diploma thesis is to develop software for automatic detection of waste in Python environment using convolutional neural network. The theoretical section is devoted to the problematic of recycling and waste. Thesis describes structures of convolutional neural network and its learning. The scope includes with algorithms and metrics for classification and detection of waste. The practical section describes dataset with all collected data and their augmentation, which are later used for training and optimization of model. The thesis includes algorithm YOLOv5 and its modification YOLOv5n, YOLOv5s and YOLOv5m, on which it is gradually trained and subsequently evaluated according to the selected metrics. It also includes a simple application that is easy to use for image evaluation.*

## **KEYWORDS**

*Waste, Python, Convolutional neural networks, Image processing, YOLOv5*

SEZNAM ZKRATEK .....	9
SEZNAM ZNAČEK (SYMBOLŮ, PROMĚNNÝCH A FUNKCÍ) .....	10
SEZNAM ILUSTRACÍ.....	11
SEZNAM TABULEK .....	13
1 ÚVOD .....	14
2 UMĚLÁ INTELIGENCE.....	15
2.1 STROJOVÉ UČENÍ.....	16
2.1.1 Počítačové vidění .....	16
2.2 HLUBOKÉ UČENÍ.....	17
3 ENVIRONMENTÁLNÍ PROBLEMATIKA.....	18
3.1 RECYKLACE .....	18
3.1.1 Třídění odpadu .....	19
4 SENZOROVÁ TECHNIKA .....	20
4.1 TYPY OPTICKÝCH SENZORŮ .....	20
4.1.1 CCD (Charge-Coupled Device) .....	20
4.1.2 CMOS (Complementary Metal-Oxide-Semiconductor) .....	20
4.2 KRITÉRIA KVALITY OBRAZU .....	21
4.3 POUŽITÍ .....	22
5 LOKALIZACE NALEZENÉHO OBJEKTU .....	23
5.1 KAMERY S AUTOMATICKÝM SLEDOVÁNÍM.....	23
5.1.1 Robotické kamery.....	23
5.2 VÝPOČET TRAJEKTORIE.....	23
6 KONVOLUČNÍ NEURONOVÉ SÍTĚ .....	25
6.1 NEURONOVÉ SÍTĚ.....	25
6.2 STRUKTURA KONVOLUČNÍCH NEURONOVÝCH SÍTÍ.....	26
6.2.1 Vstupní vrstva.....	26
6.2.2 Konvoluční vrstva .....	26
6.2.3 Příznaková mapa .....	27
6.2.4 Padding.....	28
6.2.5 Aktivační vrstva .....	28
6.2.6 Pooling vrstva.....	29
6.2.7 Plně propojená vrstva .....	29
6.2.8 Výstupní vrstva.....	30
6.3 FILTRY .....	30
7 UČENÍ NEURONOVÉ SÍTĚ .....	32
7.1 POSTUP .....	32
7.1.1 Trénování sítě .....	32
7.1.2 Chyba učení sítě .....	32
7.1.3 Metoda zpětné propagace.....	33
7.1.4 Optimalizace parametrů .....	33
7.1.5 Přeučení sítě .....	33
7.1.6 Předčasné ukončení .....	34
7.1.7 Dropout.....	34
7.2 METODY UČENÍ.....	34
7.2.1 Metoda nekontrolovaného učení .....	34
7.2.2 Metoda kontrolovaného učení .....	35

8	KLASIFIKACE OBJEKTŮ .....	37
8.1	DETEKCE.....	37
8.2	SEGMENTACE.....	37
8.3	METRIKY PRO HODNOCENÍ MODELŮ .....	38
8.3.1	mAP (Mean Average Precision).....	38
8.3.2	Matice záměn (Confusion Matrix) .....	38
8.3.3	IoU (Intersection over Union).....	39
8.3.4	Přesnost (Accuracy) .....	40
8.3.5	Úplnost (Recall) .....	41
8.3.6	Skóre F1 .....	42
9	ALGORITMY PRO DETEKCI OBJEKTŮ .....	43
9.1	R-CNN.....	43
9.1.1	Fast R-CNN.....	44
9.1.2	Faster R-CNN.....	44
9.2	YOLO (YOU ONLY LOOK ONCE) .....	44
9.2.1	YOLO v5.....	45
9.3	SSD.....	46
9.4	RETINANET.....	46
9.5	EFFICIENTDET .....	47
10	ZPRACOVÁNÍ DAT .....	49
10.1	PYTHON.....	49
10.2	KNIHOVNY .....	49
10.2.1	Pytorch.....	49
10.2.2	Ultralytics .....	50
10.3	OBRAZOVÁ DATA.....	51
10.4	DATASET.....	51
10.4.1	AUGMENTOVANÉ DATASETY .....	52
10.5	TRÉNOVÁNÍ YOLO V5.....	54
10.5.1	TRÉNOVACÍ SKRIPT .....	54
10.5.2	OPTIMALIZAČNÍ METODY.....	56
10.6	TRÉNOVANÉ MODELKY .....	57
10.6.1	YOLOv5n.....	57
10.6.2	YOLOv5s .....	59
10.6.3	YOLOv5m.....	61
10.6.4	YOLOv5l a YOLOv5x.....	63
10.7	VYHODNOCENÍ DAT .....	63
11	GRAFICKÉ ZPRACOVÁNÍ APLIKACE .....	66
11.1	KNIHOVNY POUŽITÉ PRO GUI.....	68
11.1.1	Tkinter .....	68
11.1.2	PIL .....	68
12	ZÁVĚR.....	69
	LITERATURA A ZDROJE .....	70
	PŘÍLOHY.....	77

## SEZNAM ZKRATEK

GUI	Graphical User Interface
YOLO	You Only Look Once
R-CNN	Recurrent Convolutional Neural Network
SSD	Single Shot Detector
CCD	Charge-Coupled Device
CMOS	Complementary Metal-Oxide-Semiconductor
IoU	Intersection over Union
PIL	Python Imaging Librety
PET	Polyethylentereftalát
A/D	Analog-to-digital converter
ISO	International Organization for Standardization
ReLU	Rectified Linear Unit
CNN	Convolutional Neural Network
AP	Average Precision
SPPF	Spatial Pyramid Pooling Fast
CSP	Cross Stage Partial
PAN	Path Aggregation Network
SGD	Stochastic Gradient Descent
nc	Number of Classes

## SEZNAM ZNAČEK (SYMBOLŮ, PROMĚNNÝCH A FUNKCÍ)

$f$	přenosová funkce neuronu
$x$	vstupy neuronu
$w$	paměť neuronu
$w_i$	váhy neuronu
$x_i$	vstupy neuronu
$y$	výstup neuronu
$\theta$	práh

## SEZNAM ILUSTRACÍ

Obr. 1 - Využití umělé inteligence (Greengard, 2019).....	16
Obr. 2 - Vizualizace částí umělé inteligence (Geeksforgeeks, 2024).....	17
Obr. 3 - Dosažená míra recyklace a energetického využití obalových odpadů (Feedit, 2024).....	19
Obr. 4 - Rozdíl mezi CCD a CMOS senzory (Ashtari, 2023).....	21
Obr. 5 - Příklad konvoluční neuronové sítě (Blažková, 2020).....	26
Obr. 6 - Konvoluční vrstva (Blažková, 2020).....	27
Obr. 7 - Vstup pro síť písmene X (Blažková, 2020).....	27
Obr. 8 - Vzor, vstupní obrázek a jejich rozdíl (Blažková, 2020).....	28
Obr. 9 - Konvoluční vrstva a padding (Blažková, 2020).....	28
Obr. 10 - Maxpool vrstva konvoluční sítě (Blažková, 2020).....	29
Obr. 11 - Plně propojená vrstva (Blažková, 2020).....	30
Obr. 12 - Vstupní matice a filtr s podobným tvarem (Blažková, 2020).....	31
Obr. 13 - Vstupní matice a filtr s nepodobným tvarem (Blažková, 2020).....	31
Obr. 14 - Metoda zpětné propagace (Blažková, 2020).....	33
Obr. 15 - Kohonenova síť (Blažková, 2020).....	35
Obr. 16 - Porovnání typů segmentace (Arnab, 2018).....	38
Obr. 17 - Vizualizace matice záměn (Shah, 2022).....	39
Obr. 18 - Rozdíl míry přesnosti na objektu (Shah, 2023).....	39
Obr. 19 - Rozdíl míry přesnosti (Rosebrock, 2016).....	40
Obr. 20 - Rozdíl mezi tříděním a detekcí (Rizzoli, 2021).....	43
Obr. 21 - Vizualizace YOLO algoritmu (Geeksforgeeks, 2022).....	45
Obr. 22 - Srovnání různých algoritmů (Solawetz, 2020).....	46
Obr. 23 - Detekce velkého množství objektů (Bochkovskiy, 2020).....	48
Obr. 24 - Kamera GoPro Hero 11 (GoPro, 2024).....	51
Obr. 25 - Prostředí programu makesence.ai.....	52
Obr. 26 - Ukázka výsledného formátu YOLOv5 tříd.....	52
Obr. 27 - Vizualizace datové augmentace (Kumar, 2019).....	53
Obr. 28 - Augmentace na pořízeném snímku.....	53
Obr. 29 - Spuštění trénovacího skriptu.....	54
Obr. 30 - Výsledný .yaml soubor.....	56
Obr. 31 - Průběh ztrátové funkce na trénovacích a validačních objektech YOLOv5n.....	58

Obr. 32 - Průběh ztrátové funkce na trénovacích a validačních objektech YOLOv5n .....	59
Obr. 33 - Průběh ztrátové funkce na trénovacích a validačních objektech YOLOv5s.....	60
Obr. 34 - Graf zachycující přesnost modelu YOLOv5s a chyby mAP_0.5 a mAP_0.5:0.95 .....	61
Obr. 35 - Průběh ztrátové funkce na trénovacích a validačních objektech YOLOv5m .....	62
Obr. 36 - Graf zachycující přesnost modelu YOLOv5m a chyby mAP_0.5 a mAP_0.5:0.95.....	63
Obr. 37 - Ukázka detekce .....	64
Obr. 38 - Chybovosti výsledných modelů .....	65
Obr. 39 - Graf přesností napříč modely .....	65
Obr. 40 - Ukázka aplikace .....	66
Obr. 41 - Ukázka aplikace po nahrání fotografie .....	67
Obr. 42 - Ukázka výsledné detekce plastu .....	67

## SEZNAM TABULEK

Tab. 1 - Předtrénované modely.....	57
Tab. 2 - Hodnoty prvků sítě YOLOv5n.....	58
Tab. 3 - Matice záměn modelu YOLOv5n.....	58
Tab. 4 - Hodnoty prvků sítě YOLOv5s .....	59
Tab. 5 - Matice záměn modelu YOLOv5s .....	60
Tab. 6 - Hodnoty prvků sítě YOLOv5m .....	61
Tab. 7 - Matice záměn modelu YOLOv5m.....	62
Tab. 8 - Velikosti výsledných modelů.....	63

# 1 ÚVOD

Problematika odpadů je pro životní prostředí důležitým tématem. Čím více se vynakládá úsilí na samotnou recyklaci nebo na produkty, které je možné recyklovat, tím více se chrání životní prostředí i samotný lidský život.

Umělá inteligence se v současnosti využívá na zefektivnění práce, a to zejména v průmyslu autonomních automobilů či dronů. Také se využívá ve virtuálních asistentech, vyhledávacích nebo při rozpoznávání řeči lidí, což je důležité pro zpracování obrazu. Další využití zpracování obrazu je například u detekce objektů, rozšířené reality, biometrického rozpoznávání obličejů nebo sledovacího zařízení.

Cílem této práce je sloučit tato dvě odvětví. Teoretická část pojednává o problematice odpadu a recyklace. Obsahuje základní principy neuronových sítí a jejich učení. Dále obsahuje klasifikace objektů, metriky pro hodnocení jednotlivých modelů a popis detekčních algoritmů.

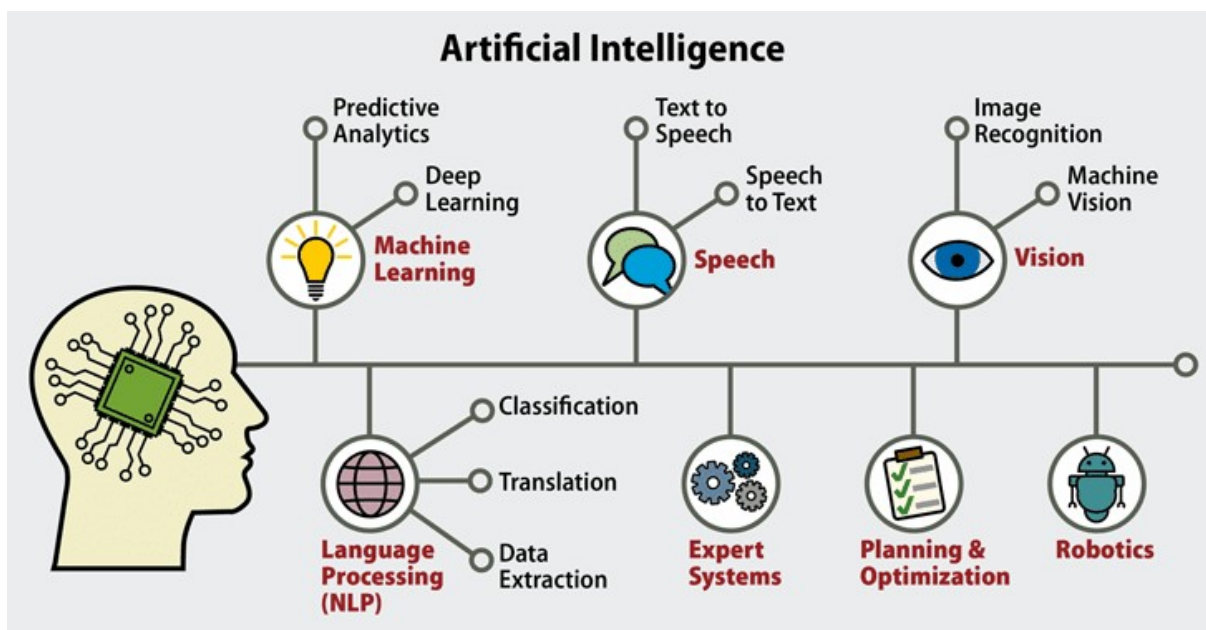
Praktická část se poté zabývá popisem datasetu a shromážděných dat. Tato data budou použita k následnému trénování a optimalizaci modelů. Součástí je také zvolený detekční algoritmus YOLO a jeho modifikace YOLOv5n, YOLOv5s a YOLOv5m. Tyto modely jsou testovány na datech a výsledky jsou následně dle zvolených metrik vyhodnoceny. Tyto data jsou porovnány mezi sebou na základě výsledků a míry chybovosti. Vyhodnocení jednotlivých snímků lze pomocí jednoduché aplikace.

## 2 UMĚLÁ INTELIGENCE

Jedná se o vědní obor zabývající se vytvářením strojů či programů, které vykazují vlastní inteligenci k nahrazení lidské práce či jejímu usnadnění. V dnešní době se jedná o jedno z nejvíce se rapidně rozvíjejících odvětví technologií.

Je to tedy počítačový systém, který s využitím matematiky a logiky dokáže napodobovat lidský proces učení a rozhodování na základě poskytnutých vzorců a dat. Přesně definována ovšem samotná umělá inteligence zatím není. Jedná se stále o velice abstraktní vědu, kde je cílem naučit stroj uvažovat jako člověk, což je samo o sobě velice komplexní úkol a zdaleka se nedá říct, že by toho bylo již dosaženo. Ovšem pro jednodušší úkony, které se za dobu vývoje dokázaly zdokonalit, se jedná o nenahraditelného pomocníka, bez kterého by se řada jiných oborů a odvětví dnes již neobešla. V konkrétních ukázkách použití je potřeba rozlišit typy umělé inteligence. Softwarová se využívá především ve virtuálních asistentech, vyhledávacích a chatech. Dále také pro rozpoznání řeči lidí, a co je důležité pro tuto práci, i pro analýzu obrazu. Druhým typem je zabudovaná neboli hardwarová. Tu lze pozorovat v rozvíjejícím se průmyslu autonomních aut a dronů, či kterýmkoli druhem robotů (Greengard, 2019).

Lze tedy říci, že umělá inteligence je nepostradatelným pomocníkem člověka i při běžných úkonech každodenního života. Usnadní různé typy úkolů, komunikaci a je, až na výjimky, neustále k dispozici. Avšak hlavní nevýhodou může být její vlastní vývoj a nepřipravenost lidí na její přítomnost. Je třeba si klást otázky, co se stane, až začne být výkonnější, než lidský pracovník a v nejhorším případě sama uvažovat, či měnit svoje naprogramování. V tomto případě se jedná o tzv. obecnou umělou inteligenci, či až umělou super-inteligenci. To jsou termíny pro tak vyspělý systém, který by byl schopen překonat lidi v naprosté většině oblastí, včetně tvořivosti a sociálních interakcí (Greengard, 2019).



Obr. 1 - Využití umělé inteligence (Greengard, 2019)

## 2.1 STROJOVÉ UČENÍ

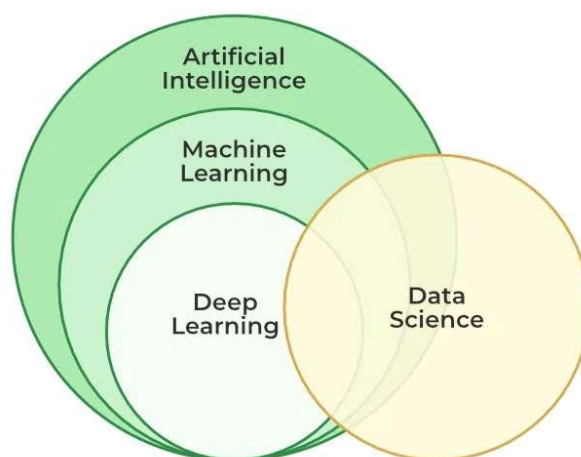
Strojovým učením nazýváme postup, kterým se počítače učí a zdokonalují, aby mohly dosáhnout umělé inteligence. Princip spočívá v nalézání vzorů za pomoci předem daných algoritmů. Z toho následně vznikají datové modely schopné predikovat budoucí výsledky. Modely strojového učení je nutné trénovat, aby se dokázaly zdokonalovat a poskytnout přesnější odpovědi. K tomu se využívají tzv. podmnožiny dat (Control Engineering Česko, 2019).

### 2.1.1 Počítačové vidění

Počítačové vidění je odvětví strojového učení. Zabývá se zpracováním dat získaných z digitálních kamer, tedy z obrázku nebo videa, ze kterých se snaží získat nejrelevantnější informace. Využívá se zejména v průmyslových a výrobních prostředích pro automatizovanou kontrolu a analýzu objektů pomocí kamery, která pořídí a zpracuje daný snímek. Kamera může být naprogramována tak, aby analyzovala pozici určitého objektu, jako je jeho poloha, tvar, velikost, barva, nebo zda je objekt přítomen na snímku a následně se podle toho vygeneruje signál. Dále se počítačové vidění využívá například pro klasifikaci, testování, sběr dat, monitorování a další. Mezi čtyři hlavní komponenty počítačového vidění patří objektivy s osvětlením, kamera, procesor a komunikační metoda (Geeksforgeeks, 2024).

## 2.2 HLUBOKÉ UČENÍ

Jedná se o pokročilý typ strojového učení využívající neurální sítě. Tak nazýváme sítě algoritmů, které jsou podobné struktuře lidského mozku. Celý proces funguje na principu návaznosti několika sad otázek na sebe, kdy s každou novou položenou se nám odemykají sady dalších souvisejících. Díky své komplexnosti se využívá v těch složitějších úkonech, než by například stačilo použít strojové učení. Jedná se tedy především o identifikaci objektů v obrazu, či k trénování umělé inteligence výkonné tak moc, že by dokázala překonat lidské myšlení. To ovšem přichází i se svou cenou, a to náročností na učení samotné. K trénování hlubokého učení je zapotřebí enormní sada dat, aby dokázalo pokrýt i různé nepřesnosti mělkého učení (Geeksforgeeks, 2024).



Obr. 2 - Vizualizace částí umělé inteligence (Geeksforgeeks, 2024)

### **3 ENVIRONMENTÁLNÍ PROBLEMATIKA**

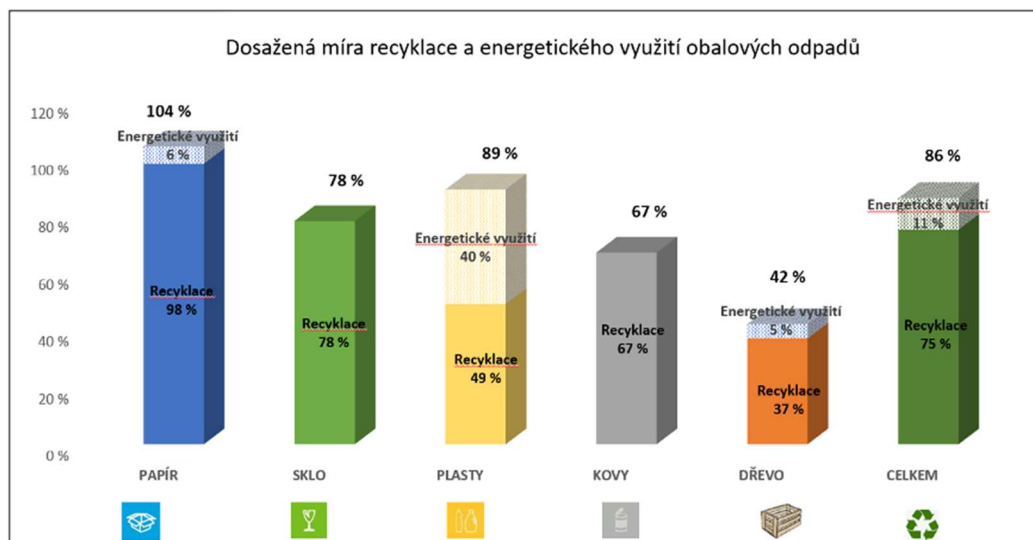
Odpadem nazýváme produkt, kterého se člověk nebo podnik vzdává z důvodu absence jeho dalšího využití. Pokud nastane tato situace, je potřeba v ideálním případě rozhodnout, jak se s ním dál naloží. V dnešní době je k dispozici recyklace (tzv. materiálové využití), anebo spalování (tzv. energetické využití). Pokud se nedá odpad ani jednou z těchto forem zpracovat, zpravidla končí na skládce odpadů, které ale nejsou nekonečné. Je tedy zapotřebí, aby každý člověk držel v podvědomí nutnost omezení vzniku odpadů, anebo užívání výrobků s dlouhou životností, aby se zamezilo dalšímu plnění již teď enormních skládek, a to už na zemi nebo ve vodě. Především je důležité brát na vědomí vznik nebezpečného odpadu, mezi který patří: oleje, tuky, kyseliny, chemikálie, zářivky, baterie, barvy, lepidla, léky, velké spotřebiče atd. (European environment agency, 2024).

Problematika odpadů je z environmentálního hlediska velice důležitým tématem. Jedná se o jeden z nástrojů, kterým člověk působí na složky životního prostředí. Odpady vznikají při veškeré, pro lidstvo prospěšné, činnosti. Jejich negativní vlivy se postupně projevují přímo na člověku, například na zdravotních problémech, ale i na ekonomických aktivitách, kdy je čím dál více zapotřebí vynášet větší náklady na vytvoření produktů, které je možné recyklovat, nebo které ideálně vůbec neznečistí prostředí (European environment agency, 2024).

#### **3.1 RECYKLACE**

Jedná se o jedno z řešení materiálového využití. Velkou výhodou je důraz na energetickou náročnost, kdy recyklace jednoho produktu stojí daleko méně než znovu vytvoření toho samého. Do této kategorie lze zařadit hliníkové plechovky, elektrochemické články, PET lahve a papír (Feedit, 2024).

### 3.1.1 Třídění odpadu



Obr. 3 - Dosažená míra recyklace a energetického využití obalových odpadů (Feedit, 2024)

V České republice je většina komunálního odpadu tvořena obaly, a to v množství 1,31 milion tun za rok, z něhož kolem 80 % je možno použít pro další zpracování. To odpovídá zhruba 1 milionu tun odpadu. Třídít lze papír, plast, sklo, bioodpad, kov, elektroniku, nápojové kartony a další. Nejvyšší míra recyklace je dosažena u papíru, a to až 90 %. Následuje potom sklo, kov, a dále plast se zhruba 45 % recyklace obalů. Průměrně tedy každý občan vytrídí zhruba 78 kg odpadu ročně (Feedit, 2024).

## 4 SENZOROVÁ TECHNIKA

Optické snímače neboli senzory pro kvalitu obrazu jsou klíčové komponenty v mnoha aplikacích, od digitální fotografie a videa po průmyslovou inspekci, zabezpečení a medicínské zobrazovací systémy. Tyto senzory přeměňují světelné paprsky na elektrické signály, které lze zpracovat a analyzovat za účelem získání podrobných obrazů, nebo pro vyhodnocení kvality obrazu.

### 4.1 TYPY OPTICKÝCH SENZORŮ

V současných fotoaparátech a kamerách se hojně využívají především dva typy snímačů. Oba fungují tak, že sbírají jednotlivé světelné částice (fotony) dopadající na jednotlivé obrazové body (pixely). Liší se tedy převážně způsobem jejich výroby a v metodách sbírání dat na pixelech.

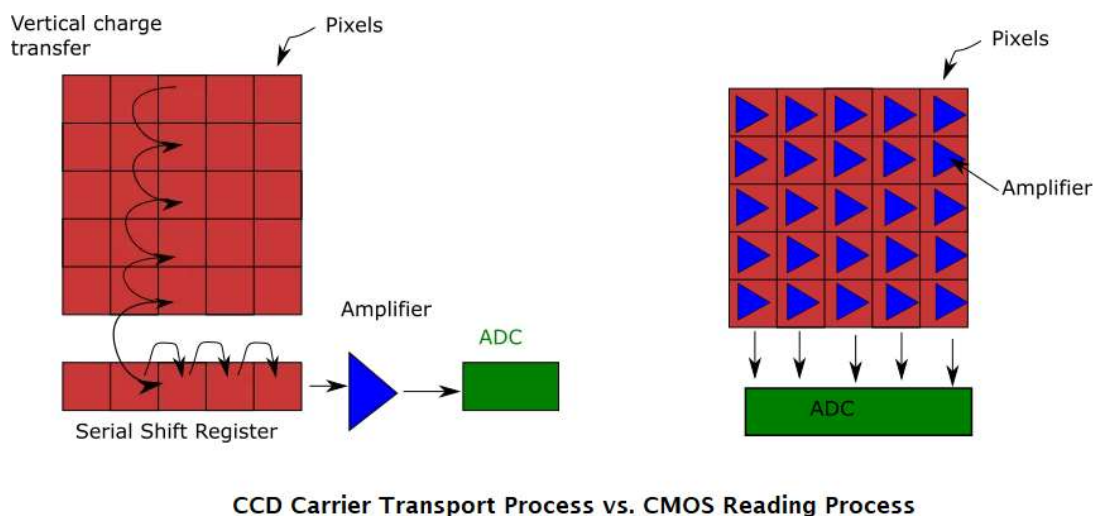
#### 4.1.1 CCD (Charge-Coupled Device)

CCD je typ optických senzorů, který je hojně využíván v profesionálních kamerách. Tyto senzory se skládají z řady fotodiod s vlastními zesilovači, které zachytávají světelné signály a převádějí je na elektrický signál. Jsou používány především pro jejich vysokou kvalitu obrazu, velmi kvalitní světelnou citlivost a nízký šum, nicméně jejich výroba není levná a mají vysokou spotřebu kvůli nutnosti samostatných čipů. Fungují na principu převodu analogového signálu jedné buňky napříč dalšími buňkami, a to vše bez ztráty kvality signálu. Jinak se tento proces nazývá fotoefekt. V moment, kdy se signál dostává k posuvnému registru na okraji snímače, dochází k převodu do zesilovače a následně do A/D převodníku. Principu, který umožňuje signálu tento pohyb, se říká nábojová vazba buněk (Ashtari, 2023).

#### 4.1.2 CMOS (Complementary Metal-Oxide-Semiconductor)

Tyto senzory používají méně energie než CCD, umožňují rychlejší zpracování obrazu a snižují celkové náklady na systém. Jejich nevýhodou ovšem vždy byla nižší kvalita obrazu, nicméně s pokrokem v technologiích se tento neduh rapidně vyrovnává. Naopak jsou velmi odolné proti obrazovému šumu. S jejich snadnější výrobou úzce souvisí lepší integrace vyššího počtu pixelů, či zesilovače nebo dalších součástí na jeden čip. Rozdíl oproti CCD senzorům je zde takový, že každá fotodioda má vlastní tranzistor, což má za následek možnost zpracování obrazu přímo na tom samém čipu. Je však také potřeba pro každý pixel

přidat mikroobjektiv, který řeší problém s nároky na velikost a potřeby vyššího zesílení (Ashtari, 2023).



Obr. 4 - Rozdíl mezi CCD a CMOS senzory (Ashtari, 2023)

## 4.2 KRITÉRIA KVALITY OBRAZU

Při posuzování kvality obrazu získaného za pomoci optických senzorů se hodnotí několik klíčových aspektů. Zde je příklad několika z nich:

### Rozlišení

Udává kolik detailů může samotný senzor zachytit. Vyšší rozlišení znamená více pixelů, a tím pádem lepší detaily obrazu (Abdulla, 2021).

### Dynamický rozsah

Jedná se o schopnost senzoru zachytit detaily i v nejsvětlejších a nejtmařších oblastech. Znamená to tedy, že senzory lépe zachytí detaily ve scénách s vyšším kontrastem. (Dolejší, 2007).

### Citlivost na světlo

Jindy také uváděna jako ISO, určuje míru schopnosti senzoru zachytit obraz i v nízkém osvětlení. Čím vyšší hodnoty ISO, tím víc lze snížit osvětlení, nicméně zároveň s tím roste i šum obrazu (Gibes, 2023).

### **Barevná reprodukce**

Toto kritérium uvádí, s jakou přesností dokáže senzor přesně zachytit barvy snímané scény. Zvolené filtry, jejich kvalita a typ tuto skutečnost napřímou ovlivňují (Wikipedia, 2024).

### **Šum**

Jedná se o velký neduh, který vzniká při nízkém osvětlení a dokáže velice snížit kvalitu obrazu. Pro jeho omezení je důležité zvolit více kvalitní senzory (Abdulla, 2021).

## **4.3 POUŽITÍ**

Optické senzory se používají především v odvětvích digitálního fotografování, průmyslové inspekce (vady na výrobních linkách), zabezpečení, dohledu (identifikace osob či vozidel) a medicinského zobrazování (například v rentgenových technologiích).

## 5 LOKALIZACE NALEZENÉHO OBJEKTU

### 5.1 KAMERY S AUTOMATICKÝM SLEDOVÁNÍM

Kamery s automatickým sledováním objektů jsou navrženy tak, aby neustále sledovaly pohybující se objekty či osoby. Kamery používají pokročilé algoritmy pro detekování a sledování pohybujících se objektů v aktuálním čase. Největší využití mají například při videokonferencích, nebo u sportovních zápasů případně v bezpečnostním monitorování.

Hlavními funkcemi automatických kamer jsou otáčení, naklánění a přibližování, aby objekt zůstal v zorném poli co nejlépe viditelný a ostrý. Některé kamery využívají infračervené, nebo ultrazvukové senzory k mapování objektu. Jiné modely kamer mohou mít funkce jako rozpoznání obličeje, případně mapování gest, jako je například přihlásit se o slovo (Zhao et al., 2024).

Lze je využít pro mnoho prostředí – od poslucháren, konferenčních místností až po bezpečnostní sledování, kde mohou poměrně rychle identifikovat narušitele. Například na sportovních akcích zlepšuje zážitek ze sledování.

#### 5.1.1 Robotické kamery

Robotické kamery jsou pokročilá zařízení, která duplikují schopnosti kamerového systému s automatickým pohybem a řízením. Používají se v mnoha odvětvích jako jsou průmyslová automatizace, medicína, bezpečnost a sledování a výzkum.

### 5.2 VÝPOČET TRAJEKTORIE

Pro správnou funkci robota je nutné také řešení jeho pohybu k detekovanému objektu. Nejdříve je ale nutné zjistit jeho přibližnou polohu  $(x_o, y_o)$  vůči poloze robota  $(x_r, y_r)$ . Z těchto hodnot se následně vypočítá rozdíl jejich odečtem:

$$\Delta x = x_o - x_r$$

$$\Delta y = y_o - y_r$$

S využitím Pythagorovy věty lze snadno dále také dopočítat vzdálenost  $d$  mezi robotem a detekovaným objektem a úhel  $\theta$  mezi osou  $x$  směrem k objektu:

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2}$$

$$\theta = \tan^{-1}(\Delta y, \Delta x)$$

Důležitý bude především výpočet trajektorie, po které se bude robot pohybovat směrem k objektu. Parametricky lze vyjádřit:

$$x(t) = x_r + \frac{t}{d} \times \Delta x$$

$$y(t) = y_r + \frac{t}{d} \times \Delta y$$

Kde  $t$  je čas potřebný k dosažení objektu (Nahavandi et al., 2022).

## 6 KONVOLUČNÍ NEURONOVÉ SÍTĚ

Jedná se o velmi specifický typ neuronových sítí zpracovávající data s maticovou strukturou, mezi které patří například obrazová data. Dokáží v nich rozpoznávat různé vzory a prvky, jako jsou okraje, tvary, barvy, textury atd. Díky svojí architektuře mohou konvoluční sítě efektivně spravovat velké objemy dat s relativně nízkým počtem potřebných parametrů, což usnadňuje jejich učení. Mají za úkol naučit se rozpoznávat objekty bez ohledu na jejich orientaci, velikost nebo polohu. Využívají pro to matematickou operaci zvanou konvoluce (Goodfellow, 2016; Institut biostatistiky a analýz Lékařské fakulty Masarykovy univerzity, 2020).

### 6.1 NEURONOVÉ SÍTĚ

Neuronové sítě jsou inspirovány principy, které můžeme pozorovat v přírodě. Základním stavebním kamenem jak přirozených, tak umělých neuronových sítí je neuron. Neurony komunikují mezi sebou prostřednictvím signálů, a v rámci neuronové sítě každý neuron přijímá několik vstupů, ale produkuje pouze jeden výstup, který může být dále předáván ostatním neuronům (QCExpert, 2020).

Výpočtem lze síť vyjádřit vynásobením hodnot přijmutých vstupů s hodnotou vah specifických pro každý neuron. Vytvoří se suma těchto hodnot a pokud je výsledné číslo větší než práh, který byl stanoven, tak dojde k transformaci již dopředu stanovenou přenosovou funkcí a to celé bude výstupem.

$$y = f\left(\sum_{i=1}^n w_i \cdot x_i - \theta\right)$$

$x_i$  – vstupy

$w_i$  – váhy

$\theta$  – práh

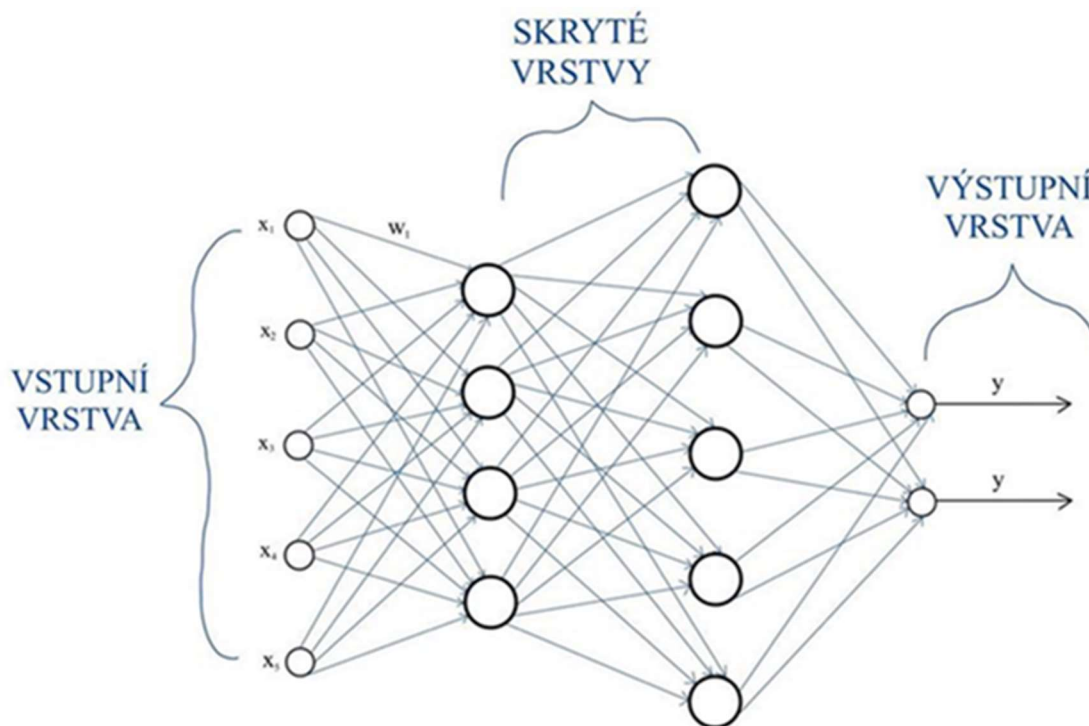
$f$  – přenosová funkce

$y$  – výstup

Vstupní část sítě přebírá a distribuuje data do následujících vrstev nazývaných skryté vrstvy. Na konci procesu, výstupní neurony, které mají jednotlivý výstup, formují výsledky zpracované celou sítí. Díky své schopnosti adaptace a učení se z poskytnutých dat, mohou neuronové sítě efektivně řešit rozličné problémy (Doležel, 2016).

## 6.2 STRUKTURA KONVOLUČNÍCH NEURONOVÝCH SÍTÍ

Konvoluční neuronové sítě jsou složeny z různých vrstev, přičemž každá má svou unikátní roli v procesu učení a zjišťování vlastností z obrazových dat:



Obr. 5 - Příklad konvoluční neuronové sítě (Blažková, 2020)

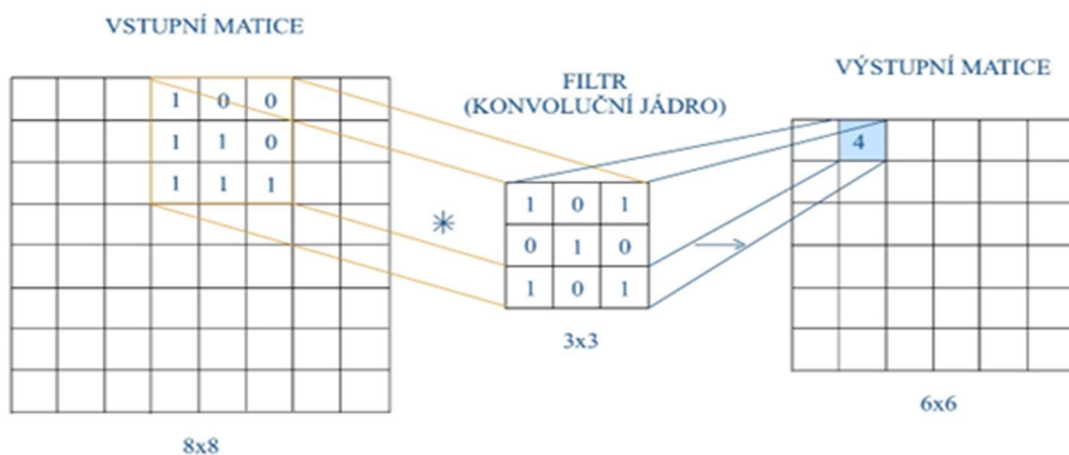
### 6.2.1 Vstupní vrstva

Začátek zpracování obrazu představuje vstupní vrstva, která přijímá matici pixelů s dimenzemi definovanými jako výška, šířka a hloubka. Tato úvodní vrstva zachycuje pixelové hodnoty z originálního snímku, které jsou dále směřovány do konvoluční vrstvy pro další zpracování (Doležel, 2016).

### 6.2.2 Konvoluční vrstva

Tyto vrstvy aplikují jádro nebo filtr, neboli matici s malým rozměrem, na vstupní data, což vede k vytvoření výstupní matice čísel (příznaková mapa). Tyto mapy umožňují síti učit se rozpoznávat vzory, jako jsou hrany, tvary, textury atd. V konvolučních vrstvách se nachází mnoho filtrů, kde každý má za úkol identifikovat specifické vzory v obrázku.

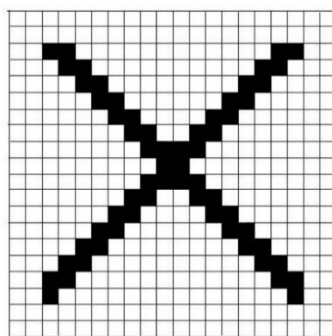
Filtry jsou učeny automaticky během procesu trénování, avšak je třeba dbát na zvyšující se počet parametrů s vysokým počtem filtrů (Karn, 2016).



Obr. 15 - Konvoluční vrstva

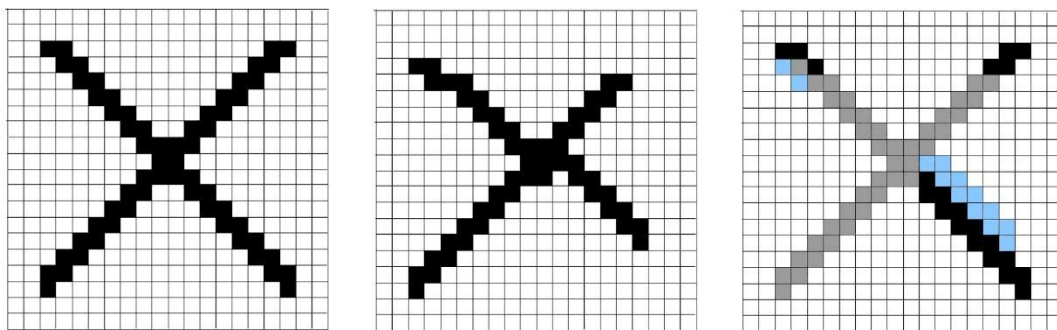
Obr. 6 - Konvoluční vrstva (Blažková, 2020)

### 6.2.3 Příznaková mapa



Obr. 7 - Vstup pro síť písmene X (Blažková, 2020)

Konvoluční vrstva se skládá z několika příznakových map. Používá se na rozpoznání znaku či symbolu ze snímku. Je ideální, aby oříznutá políčka měla stejnou velikost, proto je zde použita velikost  $32 \times 32$ .

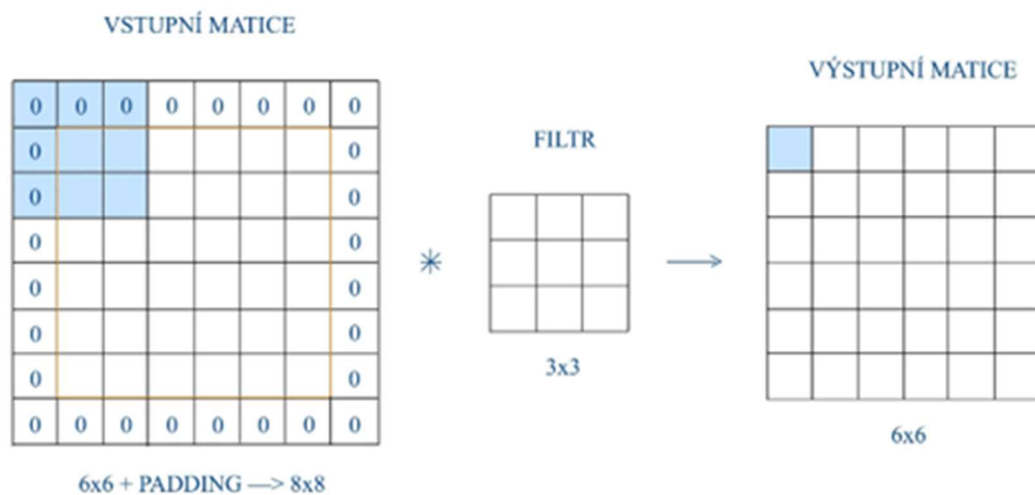


Obr. 8 - Vzor, vstupní obrázek a jejich rozdíl (Blažková, 2020)

Principem je použít ideální symbol jako vzor, poté různě mírně deformovaný, ale přesto stejný symbol jako vstupní obrázek a zaznamenávat rozdíl, kde se pixely obou symbolů neshodují (Saha, 2018).

### 6.2.4 Padding

Padding si lze představit jako umělé zvětšení matice na vstupu tím, že se ohraničí nulovými hodnotami. Tento proces je důležitý kvůli opatření proti ztrátám krajních hodnot a zároveň k ochraně původní velikosti vstupní matice (Karn, 2016).



Obr. 9 - Konvoluční vrstva a padding (Blažková, 2020)

### 6.2.5 Aktivační vrstva

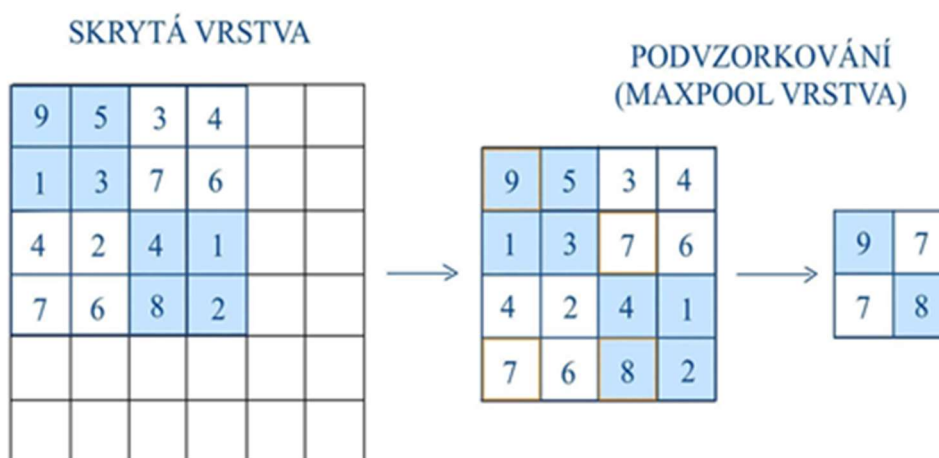
Tato vrstva může být také využita k propojení dvou odlišných neuronových sítí a může být i součástí konvoluční vrstvy. Nejčastěji se používá ReLU (Rectified Linear Unit)

aktivace, jež přidává nelinearitu do modelu. To umožňuje síti učit se komplexnější vzory. Aktivační vrstva je na každém pixelu v příznakové mapě aplikována zvlášť (Karn, 2016).

### 6.2.6 Pooling vrstva

Pooling vrstva, nebo také podvzorkování či sdružování, snižuje rozměry příznakových map. To pomáhá k snížení parametrů a přímo ovlivňuje výpočetní náročnost, která klesá zároveň s tím. Existuje několik typů sdružování. Za zmínku stojí především maxpool vrstva.

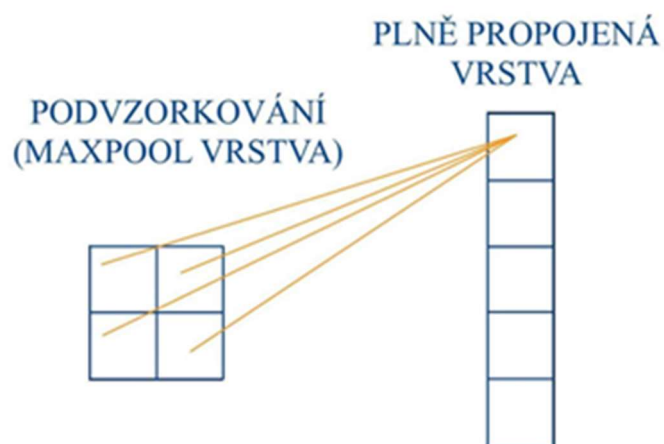
Maxpool vrstva funguje na principu, že z určité oblasti vstupu vybere nejvyšší hodnotu, kterou následně přeneseme do výstupu, jenž se posouvá k další oblasti (Moujahid, 2016).



Obr. 10 - Maxpool vrstva konvoluční sítě (Blažková, 2020)

### 6.2.7 Plně propojená vrstva

Další z vrstev je vrstva plně propojená, kdy je vytvořeno spojení mezi každým neuronem v jedné vrstvě a každým neuronem v jiné vrstvě. Pro každou vrstvu je potřeba definovat rozměr vstupu a výstupu a na vstupu je potřeba mít vektor (Karn, 2016).



Obr. 11 - Plně propojená vrstva (Blažková, 2020)

### 6.2.8 Výstupní vrstva

Poslední z vrstev je výstupní vrstva. Počet klasifikačních tříd a neuronů je zde stejný a vrstva je zcela propojena s vrstvou, která ji předchází.

## 6.3 FILTRY

Nastavení filtrů je rozuměno nastavení velikosti matice a její počet. Nejvhodnější volbou je čtvercový tvar filtrů. Určení těchto parametrů pro všechny filtry se provádí s ohledem na to, že každý filtr bude aplikován na více místech v rámci obrázku, což je možné díky přítomnosti opakujících se vzorů v obrázku. Pokud je zvolena menší velikost filtru, budou v první konvoluční vrstvě detekovány jemnější tvary (Karn, 2016).

Na obr. 12 je vstupní matice a filtr velmi podobný tvarem, to je zřejmé z toho, že se hodnoty filtru značně shodují se vstupní maticí. Při vynásobení shodných hodnot se získá vysoké číslo váhy.

VSTUPNÍ MATICE (OBRAZ)		FILTR (KONVOLUČNÍ JÁDRO)										
0	0	0	0	0	0	*	0	0	0	0	0	0
0	5	0	0	5	0		0	5	0	0	5	0
0	0	4	5	0	0		0	0	5	5	0	0
0	1	4	5	0	0		0	0	5	5	0	0
0	4	1	0	5	0		0	5	0	0	5	0
0	0	0	0	0	0		0	0	0	0	0	0
6x6							6x6					

Obr. 12 - Vstupní matice a filtr s podobným tvarem (Blažková, 2020)

Naproti tomu na obr. 13 je nepodobný tvar filtru a vstupní matice. Při vynásobení vstupní matice s filtrem se získá nízká hodnota váhy.

VSTUPNÍ MATICE (OBRAZ)		FILTR (KONVOLUČNÍ JÁDRO)										
0	0	0	0	0	0	*	0	0	0	0	0	0
1	1	0	0	0	0		0	5	0	0	5	0
0	4	0	4	0	0		0	0	5	5	0	0
0	1	1	5	0	0		0	0	5	5	0	0
0	0	1	1	0	0		0	5	0	0	5	0
0	0	0	0	0	0		0	0	0	0	0	0
6x6							6x6					

Obr. 13 - Vstupní matice a filtr s nepodobným tvarem (Blažková, 2020)

## 7 UČENÍ NEURONOVÉ SÍTĚ

Neuronové sítě lze definovat jako soustavu výpočetních jednotek, které zpracovávají data paralelně a vzájemně spolu komunikují. Tyto sítě se využívají například pro rozpoznávání a kompresi dat, tedy pro snížení velikosti obrazových, nebo zvukových souborů při zachování jejich podstatných informací. Hlavním cílem trénování neuronové sítě je optimalizovat její parametry tak, aby dosahovala co nejvyšší přesnosti ve svých výsledcích.

### 7.1 POSTUP

Učení neuronové sítě lze rozdělit na samotný proces učení, který zahrnuje trénování sítě, vyhodnocení chyby, minimalizaci chyby pomocí zpětné propagace a následnou optimalizaci. Kromě toho se zkoumají i různé jevy, které mohou během tohoto procesu nastat.

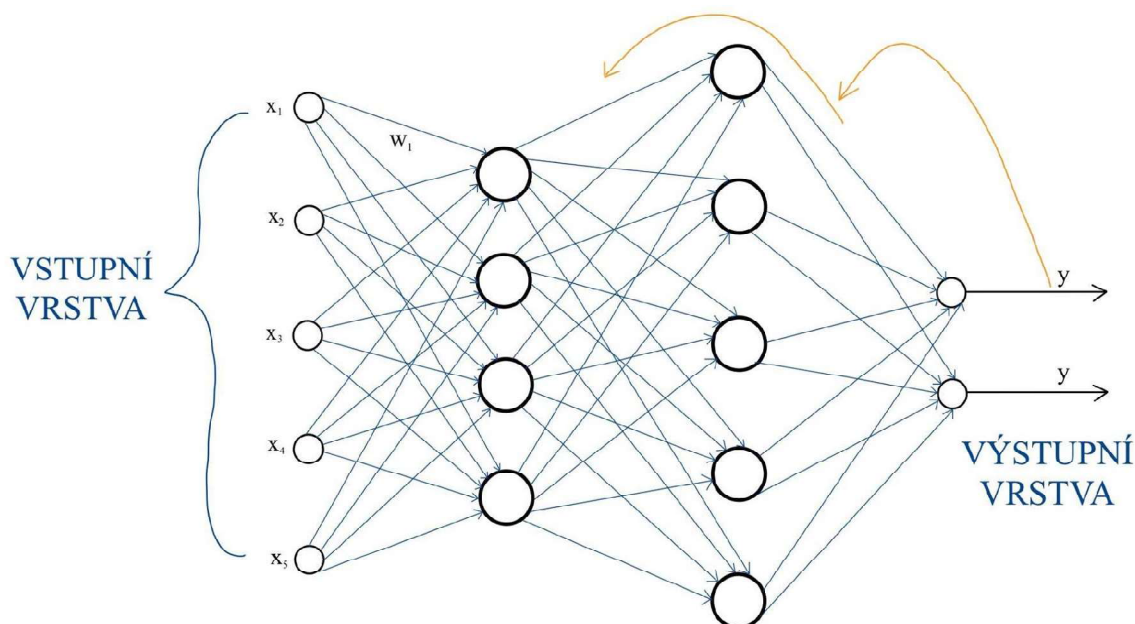
#### 7.1.1 Trénování sítě

Trénování neuronových sítí probíhá experimentálně. Cílem je optimalizace jejich vnitřních parametrů s cílem minimalizovat chybu predikce. Tento proces zahrnuje použití trénovací a testovací množiny dat. Trénovací množina slouží k samotnému učení sítě, zatímco testovací množina reprezentuje reálná data. V jednotlivých částech se počítá chyba na testovací i trénovací množině. Při výpočtu chyby na testovací množině se fáze zpětné propagace a optimalizace parametrů neprovádí. Nedostatečný nebo příliš malý počet dat může vést k přeučení sítě z důvodu nedoučení. (Feldman, 1996; Srivastava, 2014)

#### 7.1.2 Chyba učení sítě

Chybu učení lze vypočítat jako rozdíl mezi požadovaným výstupem a aktuálním výstupem sítě. Neuronová síť se učí tak, aby její generované výstupy co nejpřesněji odpovídaly požadovaným hodnotám na základě vstupních dat.

### 7.1.3 Metoda zpětné propagace



Obr. 14 - Metoda zpětné propagace (Blažková, 2020)

Tato metoda se zakládá na minimalizaci chyby, která je definována jako rozdíl mezi skutečným a požadovaným výstupem neuronové sítě, respektive jako podíl jednotlivých vah na výsledné chybě. (Vondrák, 1994)

### 7.1.4 Optimalizace parametrů

Pro snížení hodnoty chyby je nezbytné řešit optimalizaci parametrů. To zahrnuje úpravu vah tak, aby se v každém dalším kroku minimalizovala celková chyba sítě a postupně dosáhlo globálního minima (QCExpert, 2020). Nejčastěji je využívána některá z gradientních metod (Bouvier, 2006).

### 7.1.5 Přeučení sítě

Komplikací může být neúplná informace o daném průběhu učení. Informace jsou poskytovány metrikami trénovací a testovací chyby. Ideální případ by byl, kdyby obě metriky konvergovaly k nule současně. V reálné fázi učení však nastává okamžik, kdy se klesající trend testovací chyby zastaví a začne narůstat, zatímco trénovací chyba nadále klesá. Tento jev se nazývá přeučení (QCExpert, 2020).

Pokud bude na vstupu odpad, bude výstupem vektor hodnot. Síť se naučí, které hodnoty odpovídají danému symbolu a v případě, že se poté objeví nový obrázek se symbolem, síť je schopná rozeznat a dodat obrázku význam a zařadit ho do jedné ze skupin.

Výstup, který je požadovaný, může být přiřazen člověkem. V tom případě se jedná o učení se s učitelem. Pokud ale nejsou k dispozici požadované výsledky, jedná se o učení bez učitele.

### **7.1.6 Předčasné ukončení**

Nejvhodnějším způsobem, jak zamezit předčasnému ukončení tréninku, je zastavit učení v okamžiku, kdy testovací chyba začne růst. Při zastavení se uloží parametry sítě, jelikož existuje možnost, že testovací chyba opět klesne, což umožní identifikovat záchytné body po určitém počtu opakování. Následně je třeba vybrat nejlepší dosažené výsledky (Mathworks, 2016).

### **7.1.7 Dropout**

Dropout je parametr, který zabraňuje přeučení sítě. Během tréninku náhodně s určitou pravděpodobností přerušuje spojení mezi neurony. Tím dochází k průběžným změnám vah přispívajících ke ztrátové funkci, což vede k rovnoměrnějšímu rozložení vlivu vah na výslednou predikci (Srivastava, 2014).

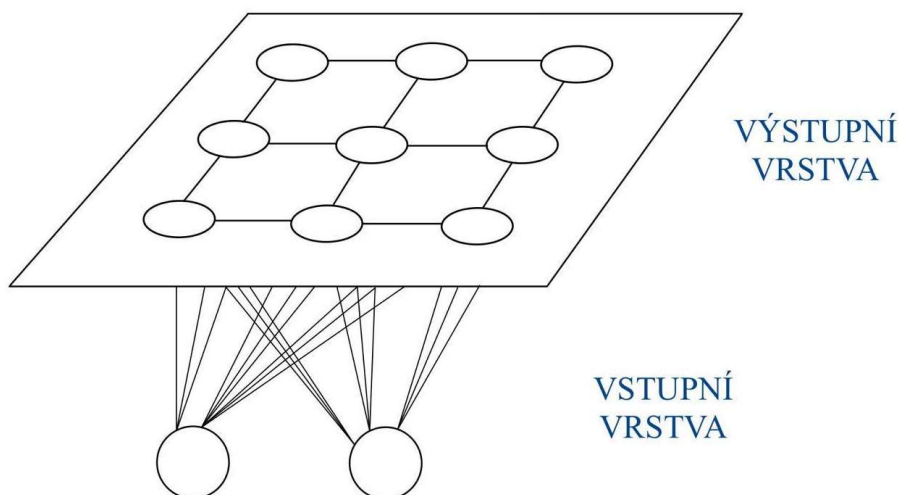
## **7.2 METODY UČENÍ**

Metody učení jsou rozděleny do dvou hlavních skupin a dalších podskupin a jejich modifikace. Mezi ty hlavní patří metody nekontrolovaného učení (také nazývané učení bez učitele) a metody kontrolovaného učení (také nazývané učení s učitelem).

### **7.2.1 Metoda nekontrolovaného učení**

Tento typ učení se také nazývá učení bez učitele. Pracuje na principu shlukové analýzy, jejímž cílem je nalézt podmnožiny neboli shluky objektů v dané množině elementů. Objekty jsou tříděny do skupin s podobnými vlastnostmi, které mohou být známy předem, ale také nemusí. V této metodě a jejím průběhu učení neexistuje žádná kontrola, zda dané učení postupuje správně. Celý proces vychází pouze z informací získaných ze vstupní množiny dat a je trvalého charakteru, protože není jisté, zda jsou výstupní data správná. Do této skupiny patří například Kohonenova síť, také známá jako Kohonenovy samoorganizační mapy (Institut biostatistiky a analýz Lékařské fakulty Masarykovy univerzity, 2020).

Kohonenova síť je určena pro rozhodování, rozlišování dat a třídění různých objektů či signálů. Je založena na principu Kohonenova učení. Mezi možné aplikace patří například hledání a detekce osob podle fotografií, zpracování obrazu, videa a fotografií, úprava zvuku, zpracování řeči a převod ručně psaného textu na tištěný text (Vojáček, 2006).



Obr. 15 - Kohonenova síť (Blažková, 2020)

### 7.2.2 Metoda kontrolovaného učení

Tento typ učení se také označuje jako učení s učitelem. Tato metoda strojového učení využívá sady trénovacích dat.

Metoda pracuje s množinou dat, která je rozdělena na dvě části, tréninkovou a testovací. Tréninková množina obvykle tvoří přibližně tři čtvrtiny celkové množiny dat a je podstatně větší, než testovací část.

Algoritmus postupně prochází jednotlivé prvky, zjišťuje odchylky od očekávaného výstupu a následně provádí korekce. K naučení určité sítě je potřeba stovky až tisíce iterací, které se nazývají epochy. Epocha představuje proces, které označuje jedno dokončené kolo tréninku modelu. Učení se nejčastěji zastaví, když je dosaženo určité hodnoty celkové chyby, což umožňuje prohlásit síť za naučenou. Učení může být také ukončeno, pokud se celková chyba ustálí a přestane klesat. Výkonnost naučené sítě se následně ověřuje pomocí testovací množiny. Nejčastějším kritériem správného naučení je střední kvadratická odchylka mezi obdrženými a očekávanými výstupy. Pokud je výkonnost na testovací množině adekvátní,

lze předpokládat, že bude obdobně dobrá i pro vstupy mimo tuto množinu (Institut biostatistiky a analýz Lékařské fakulty Masarykovy univerzity, 2020).

## 8 KLASIFIKACE OBJEKTŮ

Klasifikace objektů je nezákladnější částí strojového vidění, ve kterém se obraz zařazuje do konkrétní kategorie na základě rozpoznání objektu. Výsledkem jednoduché klasifikace je tedy právě i tato kategorie. U klasifikace s lokalizací je navíc výstupem vektor, který určuje polohu objektu na obrázku. Klasifikace objektů v počítačovém vidění zahrnuje různé přístupy, které využívají jak tradiční metody strojového učení, tak i techniky hlubokého učení (Gallagher, 2023).

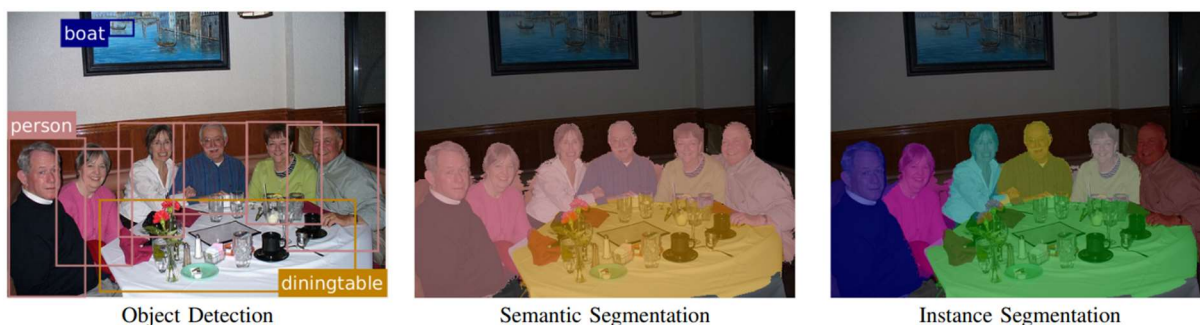
Existuje několik metod klasifikace objektů. Jednou z nich je shoda šablon, známá jako template matching. Tato metoda spočívá v porovnávání snímků s předem definovanými šablonami za účelem identifikace objektů. Další metodou je klasifikace založená na prvcích (feature-based classification). Tento přístup zahrnuje extrakci charakteristických prvků z obrazu, jako jsou hrany, rohy nebo textury, a následné využití těchto prvků k trénování modelu strojového učení pro účely klasifikace objektů. Poslední metodou je klasifikace založená na hlubokém učení. V tomto případě se používají konvoluční neuronové sítě (CNN), které se učí přímo z obrazových dat, aby rozpoznaly a klasifikovaly objekty. Díky schopnosti hlubokého učení identifikovat složité vlastnosti a vzorce v datech se tento přístup stal nejpokročilejší metodou v oblasti klasifikace objektů (Kubínek, 2009).

### 8.1 DETEKCE

Toto označení umožňuje podrobnější analýzu obrazu. Na rozdíl od klasifikace, která přiřazuje kategorii celému objektu, sémantická segmentace pracuje s jednotlivými pixely. Tento přístup rozděluje obraz na sémanticky významné části, nebo oblasti, kde každý pixel je přiřazen do odpovídající kategorie. Výsledkem je detailnější analýza obrazu. Zatímco klasifikace přiřazuje celému obrázku jednu kategorii, sémantická segmentace přisuzuje kategorie jednotlivým pixelům, a nikoli celým objektům (Kubínek, 2009).

### 8.2 SEGMENTACE

Instance segmentace se liší tím, že jde do větší hloubky obrazu než sémantická segmentace. Zatímco sémantická segmentace přiřazuje pixely kategorie bez rozlišení jednotlivých objektů stejné třídy (například všechny osoby jsou označeny stejně), instalační segmentace umožňuje rozlišovat mezi jednotlivými objekty stejné kategorie. To znamená, že každý objekt bude mít svou vlastní barvu (viz jídelní stůl, osoby obr. 16), což umožňuje identifikaci a lokalizaci jednotlivých objektů v rámci jedné třídy (Walia, 2022).



Obr. 16 - Porovnání typů segmentace (Arnab, 2018)

## 8.3 METRIKY PRO HODNOCENÍ MODELŮ

### 8.3.1 mAP (Mean Average Precision)

Mean Average Precision neboli průměrná přesnost je používána pro vyhodnocování modelů detekce, jako je například Fast R-CNN, YOLO, Mask R-CNN a další. Je založena na dalších dílčích metrikách, a to matici záměn, IoU, úplnosti a přesnosti. (Shah, 2022; Solawetz, 2020).

Metrika  $mAP$  se vypočte zjištěním průměrné přesnosti ( $AP$ ) pro každou třídu a poté zprůměrováním z několika tříd. To lze vyjádřit následujícím vzorcem, kde  $N$  je počet iterací.

$$mAP = \frac{1}{n} \sum_{i=1}^N AP_i$$

$mAP$  – průměrná přesnost

$N$  – počet iterací

$n$  – počet tříd

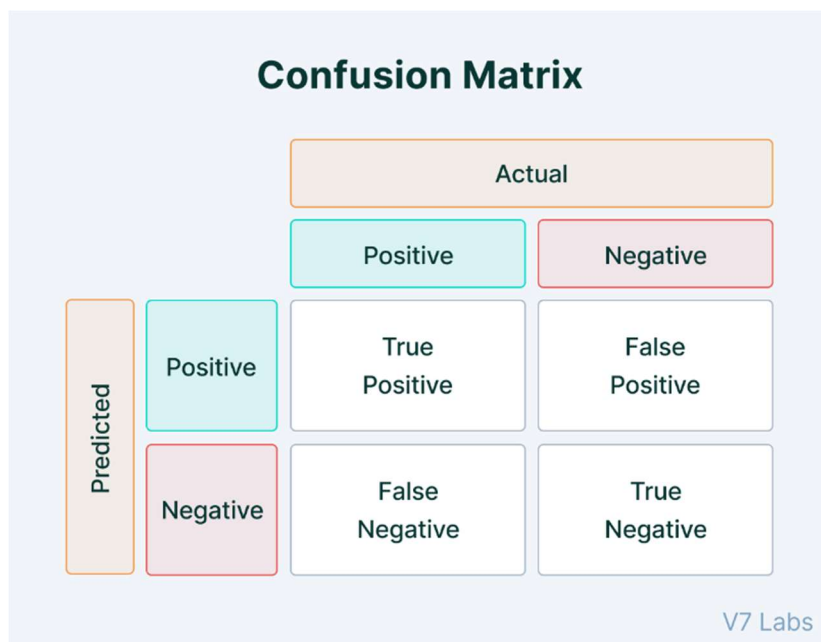
$AP_i$  – průměrná přesnost třídy  $i$

### 8.3.2 Matice záměn (Confusion Matrix)

Jedná se o tabulku používanou k hodnocení výkonu klasifikačního modelu. Její smysl spočívá v zobrazování četnosti klasifikace instancí pravdivých tříd do různých falešných tříd. Je velmi užitečná pro analýzu, nalezení chyb a zvýšení výkonu klasifikačního modelu (Kundu, 2022).

Pro vytvoření matice záměn jsou potřeba čtyři atributy, aby měla tvar čtverce. Konkrétně se jedná o pravdivě pozitivní (PP – model správně předpokládal pozitivní třídu), pravdivě negativní (PN – model nesprávně predikoval třídu), falešně pozitivní (FP – model nesprávně předpokládal třídu) a falešně negativní (FN – model správně predikoval třídu). Z této matice

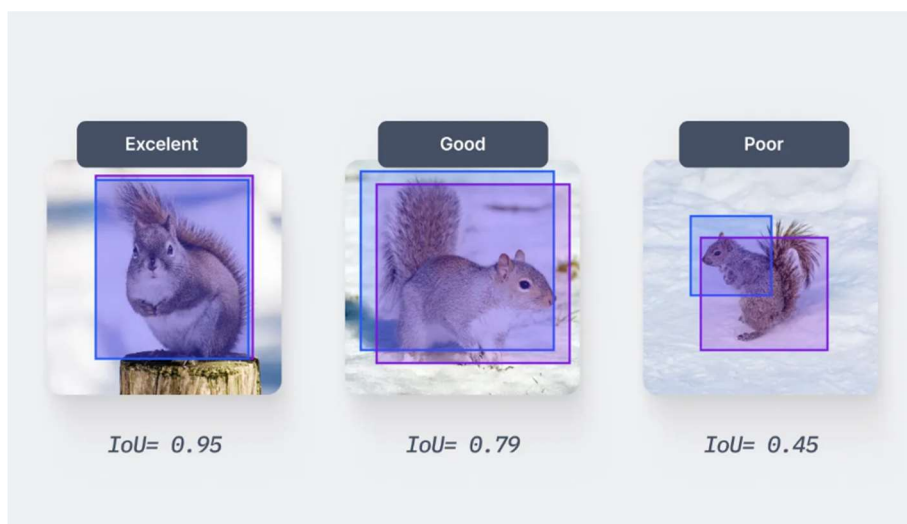
lze spočítat dále uvedené metriky pro hodnocení výkonu klasifikačního modelu (Murel, Kavlakoglu, 2024).



Obr. 17 - Vizualizace matice záměn (Shah, 2022)

### 8.3.3 IoU (Intersection over Union)

IoU je metrika hodnotící přesnost detekce a lokalizace objektů na snímcích. Měří, do jaké míry se oblast, kterou model předpovídá, překrývá s oblastí, kterou člověk označil na obrázku.



Obr. 18 - Rozdíl míry přesnosti na objektu (Shah, 2023)

Pro výpočet se spočítá průnik a sjednocení mezi oblastí označenou člověkem (A) a oblastí označenou algoritmem (B). Nakonec se IoU vypočítá jako podíl průniku a sjednocení oblastí.

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

Hodnota se pohybuje od 0 do 1, kde  $IoU = 0$  znamená žádné překrytí a 1 znamená dokonalé překrytí, proto je požadováno, aby se hodnota co nejvíce blížila k 1. Tato metrika je výhodná, protože bere v úvahu jak velikost, tak polohu ohraničujících rámečků (Shah, 2023).



Obr. 19 - Rozdíl míry přesnosti (Rosebrock, 2016)

### 8.3.4 Přesnost (Accuracy)

Přesnost měří, jak efektivně dokáže neuronová síť předpovědět správný výsledek z nových, nevyzkoušených dat.

Jako příklad lze uvést: existuje-li  $n$  obrázků, které obsahují nějaké množství  $m$  odpadu, tak po zpracování všech obrázků model nalezne 100 OD (odpadu). Z toho 90 je pravdivě pozitivních detekcí, tedy objektů, které byly správně identifikovány, jako pozitivní (PP = 90) a 10 falešně pozitivních, tedy objektů, které byly chybně označeny jako negativní (FP = 10). Přesnost tohoto modelu se vypočítá dosazením do vzorce:  $P = \frac{90}{90+10} = 0,9$ . Zjistilo se, že model má přesnost  $P = 0,9$ .

Pokud model detekuje objekt, přesnost vyjadřuje frekvenci pravdivě pozitivní detekce. Přesnost (P) modelu je poměr mezi pravdivě pozitivně detekovanými objekty a celkovým počtem objektů, které jsou detekovány. Celkový počet detekovaných objektů zahrnuje pravdivě pozitivní (PP) i falešně pozitivní (FP) detekce. To vyjadřuje vzorec:

$$P = \frac{PP}{PP + FP}$$

*P – přesnost*  
*PP – pravdivě pozitivní*  
*FP – falešně pozitivní*

Přesnost modelu se snižuje, pokud model provádí velké množství falešně pozitivních detekcí nebo má málo pravdivě pozitivních detekcí. Naopak, přesnost se zvyšuje v případě, kdy model provádí hodně pravdivě pozitivních detekcí nebo malé množství falešně pozitivních detekcí (Fränti; Mariescu-Istodor, 2023).

Pokud je  $n$  obrázků, na kterých je  $m$  objektů, po zpracování obrázků modelem vyjde to, že model našel 100 objektů. Z toho je 90 pravdivě pozitivních (PP = 90) a 10 falešně pozitivních (FP = 10). Přesnost je následně vypočítána dosazením do vzorce:

$$P = \frac{90}{90 + 10}$$

Tím se zjistí, že model má přesnost  $P = 0,9$ .

### 8.3.5 Úplnost (Recall)

Úplnost je metrika, která hodnotí, jestli model našel objekt pokaždé, když ho měl najít. Čím je větší, tím více objektů z testovací množiny model pravdivě pozitivně detekoval.

Jedná se o poměr pravdivě pozitivně detekovaných objektů k celkovému počtu objektů v testovacích datech. Celkový počet objektů je dán součtem pravdivě pozitivních (PP) a falešně negativních (FN) detekcí (Fränti; Mariescu-Istodor, 2023).

$$Ú = \frac{PP}{PP + FN}$$

*Ú – úplnost*  
*PP – pravdivě pozitivní*  
*FN – falešně negativní*

### 8.3.6 Skóre F1

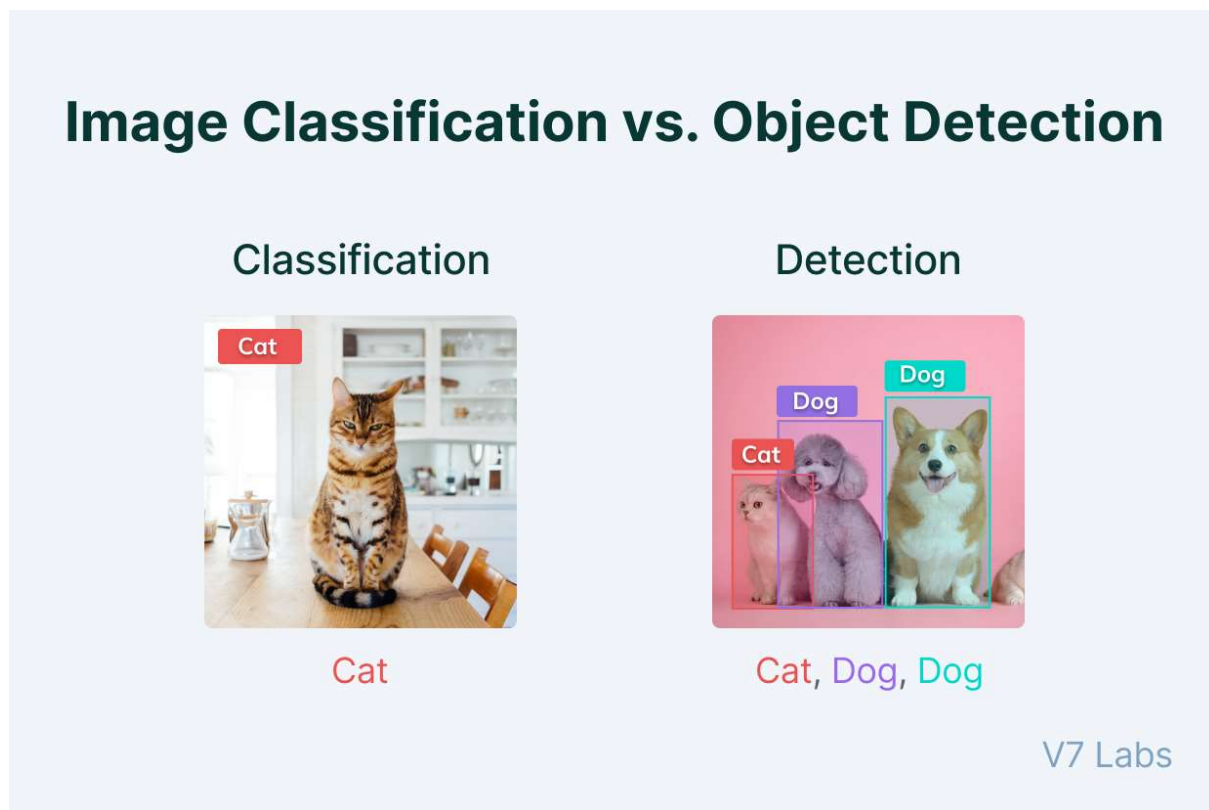
F1 skóre je metrika používaná k hodnocení výkonu klasifikačního modelu, pokud chceme vyvážit přesnost a úplnost. Je to harmonický průměr přesnosti a úplnosti definován následujícím vzorcem:

$$F1 = 2 \times \frac{P \times \bar{U}}{P + \bar{U}}$$

Tato metrika je užitečná, pokud jsou brány v potaz obě metriky, a to správnost i úplnost. Čím vyšší je skóre F1, tím je lepší výkon modelu. Maximální hodnota je 1, tedy dokonalý model, minimální je 0 (Franti; Mariescu-Istodor, 2023).

## 9 ALGORITMY PRO DETEKCI OBJEKTŮ

Algoritmus je postup nebo návod pro řešení určitého druhu problému, který je prováděn pomocí konečného množství přesně definovaných kroků. Je jím myšlen teoretický princip řešení problémů. V počítačovém vidění tvoří základ velkého množství různých aplikací a jsou do nich implementovány především pro jejich schopnost identifikovat a lokalizovat objekty v obrázcích nebo videích (Rizzoli, 2021).



Obr. 20 - Rozdíl mezi tříděním a detekcí (Rizzoli, 2021)

### 9.1 R-CNN

Tento algoritmus pracuje na principu selektivního vyhledávání potencionálních objektů ve snímku a poté provádí extrakci atributů a klasifikaci objektů s pomocí konvoluční neuronové sítě v každé vytipované oblasti. Toto selektivní vyhledávání má svá vlastní pravidla. Po selekci dvou tisíc návrhů oblastí každou z nich transformuje do čtverce s velikostí 227×227 pixelů. Ty dále propaguje do pěti konvolučních vrstev, ze kterých vzejde výstupní vektor o délce 4096. Finální vyhodnocení mají za úkol dvě plně propojené vrstvy, které poskytnou informace o oblasti a poloze vektoru v ní (Potrimba, 2023).

Algoritmus je přesný, nicméně velice časově náročný. Není ho tedy možné použít pro operace vyžadující detekci v reálném čase, protože zpracování všech dvou tisíc oblastí se provádí zvlášť. Do této rodiny algoritmů spadají další vylepšení, které stojí za zmínku.

### **9.1.1 Fast R-CNN**

Toto vylepšení má za výsledek stonásobné zvýšení rychlosti oproti R-CNN, a také daleko větší přesnost detekce. Rozdíl je zde především v tom, že se obraz zpracovává konvolučními vrstvami jako celek. To má za následek vznik příznakové mapy, jež je rozdělená na další oblasti. Před klasifikací plně propojenými vrstvami jsou zde však ještě pooling vrstvy, které transformují zmíněné oblasti na příznakové vektory fixních délek. Nicméně i přes tato vylepšení algoritmus nedosahuje takových rychlostí, aby byl vhodný pro využití v real-time systémech. To ovšem stále platí i pro jeho nástupce (Potrimba, 2023).

### **9.1.2 Faster R-CNN**

Další vylepšení představuje Faster R-CNN, které spojuje konvoluční neuronové sítě s regionálními konvolučními vrstvami (Region Proposal Network), čímž nahrazuje selektivní vyhledávání pro generování ohraničujících rámečků. V klasickém přístupu je vstupní obraz nejprve zpracován konvolučními vrstvami, které vytvářejí příznakovou mapu, a ta je následně převedena na vektor, který je analyzován plně propojenými vrstvami. V případě Faster R-CNN se však příznaková mapa dále používá jako vstup pro regionální vrstvy. Po této mapě se posouvá malé okno, které slouží jako vstup do sítě, která generuje návrhy regionů. Tyto návrhy jsou parametrizovány relativně k referenčním boxům nazývaným kotvy (anchors). Každá kotva je umístěna ve středu okna a má různá měřítka a poměry stran. Obvykle se používají tři měřítka a tři poměry stran, což vede k devíti kotvám. Návrhy regionů jsou poté zpracovány do vektorové formy a odeslány do plně propojených vrstev, které poskytují informace o pozici a třídě objektu. Klíčové je, že kotvy i funkce určující regiony jsou nezávislé na poloze okna (Geeksforgeeks, 2023).

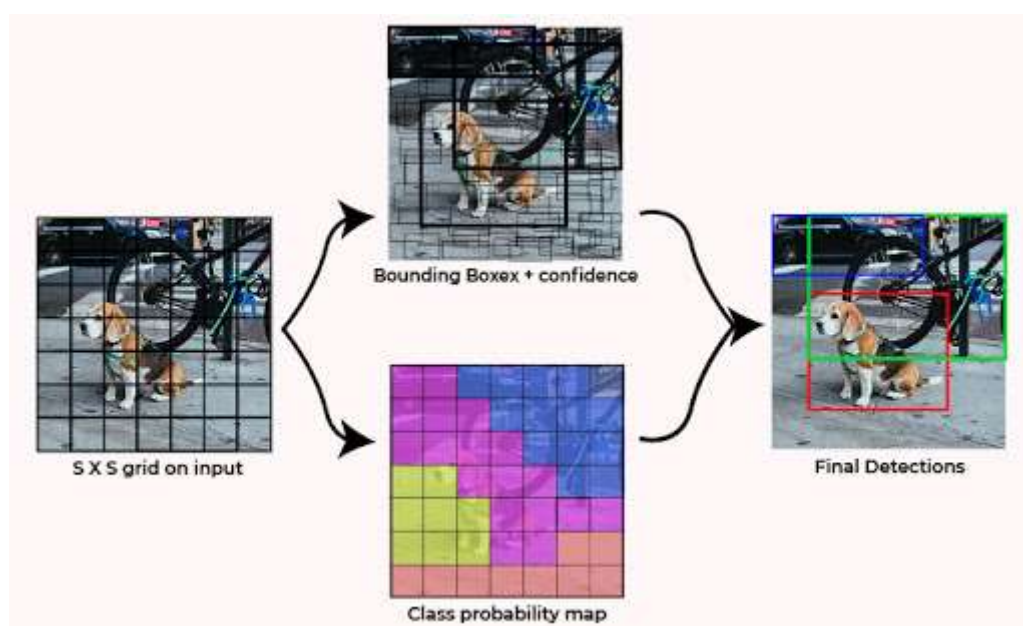
Je to velice přesný model, nicméně je pomalejší, než jeho nástupci YOLO a SSD.

## **9.2 YOLO (YOU ONLY LOOK ONCE)**

Extrémně rychlý model umožňující detekci v reálném čase. Detekuje hranice a objekty tím, že rozdělí obrázek na mřížku o velikosti  $S \times S$  za pomoci jediné end-to-end konvoluční neuronové sítě a zároveň předpovídá obsah každé buňky, uprostřed kterých se objekt

nachází. Buňky dokážou předpovídat až  $B$  rámečků (bounding boxů), které obsahují pravděpodobnostní vektor polohy objektu  $C$  pro každou třídu, a následně v případě jeho lokalizování i polohu samotnou a informaci o velikosti vztaženou k celé velikosti obrazu. Výstup sítě si lze představit jako tenzor o velikosti  $S \times S \times (B \times 5 + C)$  (Qauch et al., 2023).

Jedná se o průmyslový standard v oboru počítačového vidění a dnes se používá již verze YOLOv8. Výhodou je zde také snadné učení, protože algoritmu stačí pouze podobné obrazy, aby dokázal detekovat objekty i v trochu odlišných situacích. Mezi nevýhody naopak patří špatná detekce malých objektů, či objektů, které jsou na obrazu ve své blízkosti, či se překrývají (Kundu, 2023; Geeksforgeeks, 2022).



Obr. 21 - Vizualizace YOLO algoritmu (Geeksforgeeks, 2022)

### 9.2.1 YOLO v5

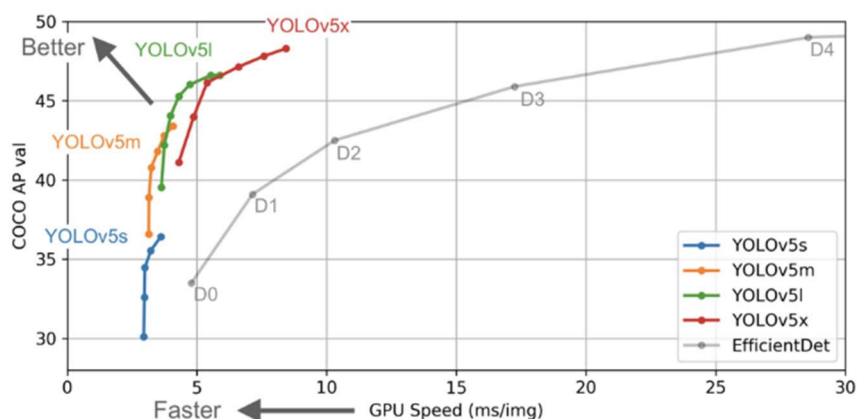
YOLOv5 je výkonný algoritmus detekce objektů, který vyvinula společnost Ultralytics. Architektura YOLOv5 se skládá ze tří hlavních částí. První je hlavní část sítě, která je navržena pomocí struktury New CSP-Darknet53, což je modifikace architektury Darknet používané v předchozích verzích. Další část je zodpovědná za generování konečného výstupu, k čemuž se využívá YOLOv3 Head. Obě tyto části spojuje struktura SPPF a New CSP-PAN (Ultralytics Inc, 2024).

YOLOv5 používá různé techniky rozšiřování dat, aby zlepšil schopnost modelu zobecňovat a omezil nadměrné přizpůsobení. Mezi tyto techniky patří například rozšíření mozaiky. Tato technika zpracování obrazu kombinuje čtyři trénovací snímky do jednoho

způsobem, který podporuje modely detekce objektů, aby se lépe vypořádaly s různými měřítky a překlady objektů. Dále je zde metoda MixUp, která vytváří složené obrazy lineární kombinací dvou obrazů a jejich přidružených značek (Ultralytics Inc, 2024; Solawetz, 2020).

YOLO a YOLOv5 patří mezi často používané systémy detekce objektů proslulé svou rychlostí, efektivitou a přesností. Tyto systémy nacházejí široké uplatnění v různých oblastech jako je robotika, autonomní vozidla a video monitorovací systémy (Kundu, 2023).

YOLOv5 obsahuje podskupiny pro detekci objektů, a to YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l a YOLOv5x.



Obr. 22 - Srovnání různých algoritmů (Solawetz, 2020)

### 9.3 SSD

Výhodou tohoto modelu je především rychlost umožňující práci v reálném čase při zachování přesnosti. Té je dosaženo principem, na kterém model pracuje. Predikce hranic a klasifikace objektů se provede na více vrstvách, což eliminuje obtíže, které byly uvedeny u algoritmu YOLO. Ve své podstatě na první pohled vypadá algoritmus SSD stejně. Obrázek prochází pouze jednou sítí (tu si lze pomyslně rozdělit na základní síť, která se soustřeďuje pouze na získání příznakové mapy a navázanou strukturu několika konvolučních vrstev, jež snižují velikost mapy) (Paul, 2020).

### 9.4 RETINANET

Tento algoritmus je založený na jeho předchůdci ResNet, ovšem zde je doplněn o tzv. Focal Loss. Jedná se o funkci ztráty, která pomáhá řešit nerovnováhu mezi pozitivními a negativními vzory při tréninku, což znamená, že se dokáže učit i na méně častých

objektech. Dosahuje rovnováhy mezi vysokou přesností a rychlostí. Za zmínku stojí architektura tohoto algoritmu. Tu tvoří tzv. páteř, která zpracovává vstupní obraz. Ta vytváří příznakové mapy a dvě podsítě pro klasifikaci objektů a vypočítání jejich polohy. Architektura páteře samotné se nazývá Feature Pyramid Network (její základ tvoří již uvedená architektura ResNet) a jak už z názvu vyplývá, modifikuje konvoluční neuronové sítě do pyramidového tvaru a spojuje je mezi sebou tak, aby v různých měřítkách došlo k lepší detekci. Předlohou pro nalezení každé oblasti je zde již uvedený Regional Proposal Network (Faster R-CNN), zde však na principu kotev, které jsou vstupem podsítí pro klasifikaci a polohu. Kotvy slouží pro výpočet pravděpodobnosti výskytu daného objektu. Co se týče podsítí pro klasifikaci, ty jsou navázány na každou z vrstev páteře ve formě konvoluční sítě. Polohová podsít' je v tomto smyslu totožná (Li, Ren, 2019).

## 9.5 EFFICIENTDET

EfficientDet je jedna z neúčinnějších dostupných metod z hlediska poměru přesnosti a rychlosti, jež kombinuje EfficientNet, jakožto efektivní architekturu konvolučních sítí, s optimalizovaným frameworkem pro detekci objektů. To umožňuje dosažení vysokých úrovní přesnosti bez složitých výpočtů při zachování kvality i na zařízeních s omezenými výpočetními prostředky. Dosahuje toho díky provedení efektivního škálování současně v rozměrech hloubky sítě, šířky a rozlišení vstupního obrazu. Je zde využita nadstavba pyramidového tvaru uvedeném v předchozím bodě, tzv. Bidirectional Feature Pyramid Network, která umožňuje efektivnější kombinaci informací z různých měřítek pro snazší detekci objektů s různými velikostmi. Samozřejmě je zde kladen velký důraz na hardwarovou efektivitu, díky níž je možný velice rychlý trénink a nízké náklady na výpočetní výkon, které jsou pro tento algoritmus zásadní (Tan et al., 2020).



Obr. 23 - Detekce velkého množství objektů (Bochkovskiy, 2020)

## 10 ZPRACOVÁNÍ DAT

Pro zhotovení praktické části byl využit programovací jazyk Python, kvůli jeho rychlosti a optimálnosti. Je v něm implementován YOLOv5, který využívá open-source Ultralytics, což je intuitivní AI platforma pro práci s Deep learning.

### 10.1 PYTHON

Python je programovací jazyk vytvořený v 80. letech 20. století, který se používá pro vývoj různých typů softwarových aplikací, ať už jde o datové vědy, automatizaci, nebo webové stránky.

Jedná se o vysoce úroňový skriptovací jazyk, který podporuje objektové programování, procedurální a funkcionální programování. Pro jeho všestrannost se používá od datové analýzy až po umělou inteligenci automatických úloh.

Mezi hlavní výhody patří snadnost naučení, rozsáhlé standartní knihovny a velmi aktivní komunita. Programovací jazyk je multiplatformní, dá se tedy použít na různých operačních systémech a integruje různé přístroje a programy. V základní instalaci se také nachází debugger, který je ideální pro svižné a velmi jednoduché ladění kódu.

Python se často používá k vývoji back-endu webových stránek nebo aplikací, tedy částí, které běžný uživatel nevidí. Odesílá data na servery a ze serverů, zpracovává data a komunikuje s databázemi. Také řídí směrování URL a zajišťuje bezpečnost. Python poskytuje několik frameworků pro vývoj webových aplikací, z nichž mezi nejpoužívanější patří Django a Flask (Staff, 2024).

### 10.2 KNIHOVNY

#### 10.2.1 Pytorch

PyTorch je open-source knihovna, která je založená na knihovně Torch, jež je určena primárně na účely strojového a hloubkového učení. Tato knihovna se nejčastěji používá v kombinaci s jazykem Python, ale lze použít i s jazykem C++. Největší využití PyTorch je v oblasti zpracování přirozeného jazyka. PyTorch je vyvíjen laboratoří Facebook AI Research. Zachovává si možnosti výpočtu tenzorů Torche, ale poskytuje více uživatelsky

přívětivé rozhraní. To vývojářům usnadňuje programování, iteraci a vytváření modelů hlubokého učení (Staff, 2024).

### **10.2.2 Ultralytics**

Knihovna Ultralytics je zaměřená na práci s obrazovými daty. Spočívá v analýze obrazových dat, segmentaci, tréninku modelů a v detekci dat.

Analýza dat zahrnuje zpracování obrazových souborů. Segmentace je proces rozdělení obrazu na jednotlivé sekce, tedy segmenty, dle určitých kritérií. Pro detekci a lokalizaci obsahuje knihovna nástroje, které jsou užitečné například pro sledování pohybu a dají se dále využít u kamer nebo autonomních zařízení (Jocher, 2023).

## 10.3 OBRAZOVÁ DATA

GoPro je americká firma založená v roce 2005. Největším úspěchem se stala digitální kamera Digital HERO 3. V roce 2014 byla vyvinuta kamera HERO 3+, která již dokázala natáčet a dělat fotografie ve 4k a měla fotoaparát o rozlišení 12Mpx.

K pořizování záznamu videí a snímků byla použita kamera GoPro Hero 11. Tato kamera bylo zvolena pro její velikost 7,2 cm × 5,0 cm a lehkost 154 g. Rozlišení kamery je poměrně vysoké, a to 27.6Mpx (5599x4927) (GoPro, 2024).



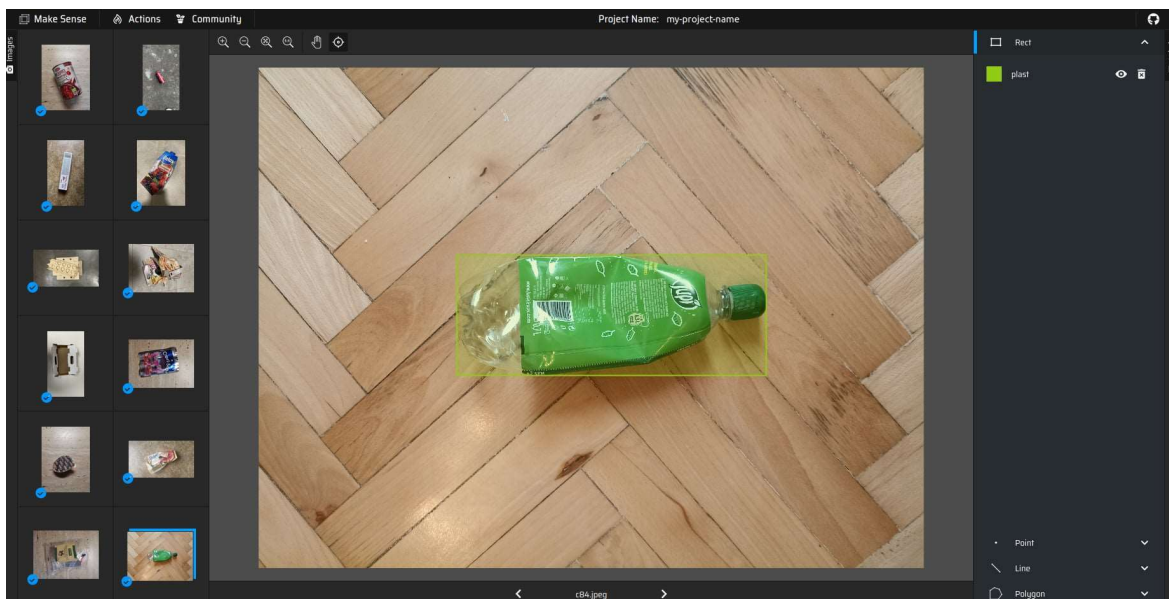
Obr. 24 - Kamera GoPro Hero 11 (GoPro, 2024)

## 10.4 DATASET

Nutností bylo nejdříve upravit dataset a zkontrolovat četnost snímků. Na určitém počtu snímků se vyskytovala pouze jedna z vrstev, a to plast, kov, papír, sklo, a na zbylých kombinace těchto vrstev.

Celkově bylo pořízeno 489 snímků, z toho 222 fotografií plastu, 67 fotografií kovu, 136 fotografií papíru a 47 fotografií skla. Kombinovaných fotografií bylo 25.

Pro zpracování obrazu a vytvoření labelů pro fotky byl zvolen open-source program makesense.ai. Jednotlivé snímky byly nahrány do programu a na každém bylo určeno, jaká vrstva se kde na snímku nachází.



Obr. 25 - Prostředí programu makesence.ai

Pro následné trénování bylo zvoleno 418 fotografií a 71 fotografií pro validaci. Po určení výskytu vrstev na snímcích byl vyexportován YOLO formát, s kterým se následně pracovalo při tréninku. YOLO formát výstupu je tabulka, ve které se nachází, jaká vrstva se vyskytuje na souřadnicích snímku. 0-3 je vrstva, která určuje, o jaký odpad se konkrétně jedná. A souřadnice  $x,y$ , které ohraničují okraje prvku na snímku.

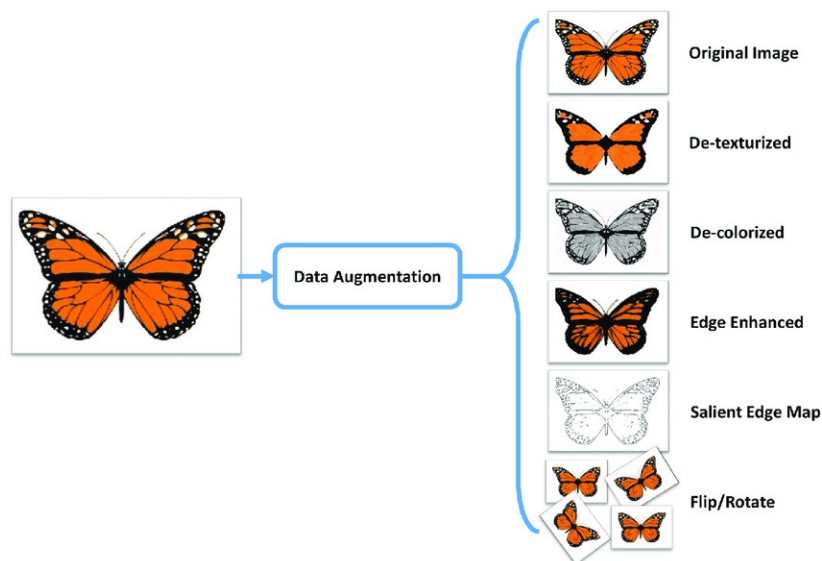
```
3 0.455954 0.407255 0.377154 0.143976
3 0.413897 0.590659 0.309320 0.202483
3 0.701963 0.531644 0.237869 0.190273
```

Obr. 26 - Ukázka výsledného formátu YOLOv5 tříd

Na obrázku číslo 25 je ukázka výsledného formátu YOLOv5 labelů pro jednotlivé snímky. 0 – papír, 1 – sklo, 2 – plast, 3 – kov.

### 10.4.1 AUGMENTOVANÉ DATASETY

Augmentace dat je technika, která se využívá k rozšíření již stávajícího datasetu obrázků nebo dat. Tato technika umožňuje z původních obrázků vygenerovat různě upravené varianty pomocí úprav a efektů, jako je například oříznutí, otočení, či různé změny barev pomocí saturace a filtru. Na každý obrázek lze aplikovat jakýkoliv počet úprav (Kumar, 2019).



Obr. 27 - Vizualizace datové augmentace (Kumar, 2019)

Geometrické augmentace zahrnují ořezávání, zrcadlení, rotace nebo změnu velikosti obrázků. Model se tak naučí rozpoznat objekty v různých polohách a velikostech.

Změna kontrastu nebo jasu přispívá k robustnosti. Inverze barev může pomoci rozpoznávat objekty na negativních snímcích.

Přidání šumu pomáhá lépe zvládat reálné podmínky, kde jsou obrázky často zašuměné.



Obr. 28 - Augmentace na pořízeném snímku

## 10.5 TRÉNOVÁNÍ YOLO V5

Pro trénovací množinu bylo zvoleno přibližně 85 % snímků a 15 % snímků pro validační množinu. Trénovací část slouží pro učení modelu k rozpoznávání objektů na snímcích. Při samotném trénování se využívá snímků a labelů ke zlepšení svých vah pomocí zpětného šíření chyby a optimalizačních algoritmů (SGD nebo ADAM).

Trénovací množina musí obsahovat větší množství snímků, aby měl model prostor naučit se různé variace a charakteristiky objektů. Validací množina slouží k hodnocení modelu při trénování a zkouší, jak dobře byla data generalizována pro nové snímky, které nebyly použity pro trénování.

Během trénování se model testuje po daných epochách na validační množině. Validací množina musí být reprezentativní vzorek celkového datasetu.

### 10.5.1 TRÉNOVACÍ SKRIPT

Trénovací skript *train.py* obsahuje parametry, které jsou potřeba nastavit pro optimální trénování. Mezi parametry patří:

--img určuje rozlišení vstupního snímku

--batch určuje počet vzorků, tedy dávku

--epochs je počet epoch

--data odkazuje na cestu, kde nalezneme soubor datové sady

--weight nastavuje výchozí váhy

```
python train.py --img 640 --batch 16 --epochs 200 --data coco128.yaml --weights yolov5s.pt
```

Obr. 29 - Spuštění trénovacího skriptu

#### Rozlišení snímků

Rozlišení snímků je vyjádřeno počtem obrazových bodů, ze kterých je následně vytvořena výsledná fotografie.

Rozlišení u displejů se udává v počtu pixelů, tedy obrazových prvků. Tato hodnota je prezentována jako dvě čísla, kde první hodnota označuje počet pixelů na šířku a druhá hodnota na výšku. Každý pixel má svou barvu a společně s ostatními tvoří obraz.

U obou těchto pojmů rozlišení platí, že čím větší rozlišení, tím lepší kvalita a ostrost.

Pro vytvoření modelu bylo nastaveno rozlišení 640×640 pixelů, jelikož je to nejvíce optimální pro balanc přesnosti a zátěže na hardware. Menší rozlišení například 416×416 může způsobovat menší přesnost a zároveň nižší zátěž na hardware a je optimálnější pro detekci menších objektů. Větší rozlišení například 1024×1024 je mnohem náročnější na hardware, ale může vzniknout větší přesnost modelu (Šimečík, 2023).

### **Batch-size**

*Batch-size* je rozdělení na určitý počet vzorků, který je následně zpracováván v jedné iteraci při trénování neuronové sítě. Výhodou tohoto řešení může být méně potřebné paměti kvůli menšímu počtu vzorků, a na základě toho probíhá i rychlejší trénink. Nevýhodou je, že čím menší bude dávka, tedy počet vzorků, tím méně bude přesný odhad gradientu (CrossValidated, 2022).

Pro trénování byla zvolena *batch-size* defaultní 16, která se zdála pro použitý software neoptimálnější (Thakur, 2022).

### **Epocha**

Pojem epocha vyjadřuje jedno dokončené kolo tréninku modelu.

Modely byly trénovány nejprve na 15 a 35 epochách, ale výsledky byly nedostačující. Následně bylo zvoleno 200 epoch, kde už se výsledné přesnosti nezvyšovaly, ale stagnovaly.

### **Data**

Tento parametr ukazuje cestu k .yaml souboru, který definuje cesty v trénovací a validační množině a zároveň ukazuje klasifikace, zda je trénovací množina ve složce train a validační množina ve složce val. Také je nastaveno *nc* neboli počet tříd a následně jejich názvy. Pro práci byl změněn soubor coco128.yaml pro potřebné účely.

```
path: ../train_data/  
train: images/train/  
val: images/val/  
  
nc: 4  
  
# Classes  
names:  
  0: papir  
  1: sklo  
  2: plast  
  3: kov
```

Obr. 30 - Výsledný .yaml soubor

## Váhy

Parametr *weights* odkazuje na cestu k předtrénovaným vahám, které jsou vhodné pro počáteční trénování modelu. Předtrénované váhy byly trénovány na datasetu *COCO*, kde jsou datasety dostupné v mnoha velikostech od varianty nano po extra large. Pro práci byly zvoleny modely nano, small a medium. Ty byly pro software ještě přijatelné. Jedna epocha u medium předtrénované váhy trvala 13 minut.

## 10.5.2 OPTIMALIZAČNÍ METODY

Optimalizační metoda ADAM, která byla v této práci zvolena jako defaultní, je adaptivní algoritmus pro učení navržený tak, aby zlepšil rychlost tréninku v hlubokých neuronových sítích a dosáhl rychlé konvergence. Mezi jeho výhody patří efektivnost na velkých datasetech nebo robustnost. Nevýhodou může být vyšší paměťová náročnost oproti SGD (Agarwal, 2023).

Oproti tomu SGD neboli Stochastic Gradient Descent je optimalizační algoritmus, který aktualizuje parametry modelu na základě gradientu ztrátové funkce. Výhodou je rychlá konvergence na malých datasetech a také jednoduchá implementace. Nevýhodou může být oscilace kolem minima na špatně podmíněných funkcích (Landro, 2020).

ADAM je častěji preferován pro složitější úlohy při práci s velkými daty, proti tomu SGD je rychlejší na menších datasetech.

## 10.6 TRÉNOVANÉ MODELKY

Algoritmus YOLOv5 nabízí předtrénované modely, tedy definované struktury sítí, které mají různou velikost, přesnost nebo rychlost učení.

Modely jsou trénované na velkém datasetu *COCO*, který je optimální na trénování vlastního datasetu. Předtrénované modely mohou velmi zlepšit kvalitu a rychlost trénování. Předtrénované modely pro YOLOv5 jsou:

Metrika  $mAP_{0.5:0.95}$  má rozsah prahových hodnot 0,5 – 0,95. Tento rozsah znamená, že se vyhodnocuje výkon daného modelu při různých úrovních překrytí vyhodnocených a referenčních rámečků. Obecně platí, že čím větší číslo, tím lepší výkon detekce objektů. Volba prahu velmi ovlivňuje výslednou detekci. Příliš nízký práh může způsobovat falešně pozitivní detekce, zatímco vysoký práh může mít za následek příliš přísná kritéria.

Model	Velikost (px)	mAP 50 – 95 (%)	mAP 50 (%)	Rychlost CPU (ms)	Rychlost GPU (ms)	Parametry (mil)
V5n	640	28	47,5	45	6,3	1,9
V5s	640	37,4	56,8	98	6,4	7,2
V5m	640	45,5	64,1	224	8,2	21,2
V5l	640	49	67,3	430	10,1	46,5
V5x	640	50,7	68,9	766	12,1	86,7

Tab. 1 - Předtrénované modely

Z těchto variant byly použity YOLOv5n, YOLOv5s, YOLOv5m.

### 10.6.1 YOLOv5n

Jedná se o nano neuronovou síť. Tato verze je nejmenší s nejrychlejším tréninkem, ale s nižší detekční schopností. Jedna epocha u tohoto modelu trvala přibližně 2 minuty a 30 sekund. Velikost obrázku byla zvolena 640px, *batch-size* 16 a počet epoch 200. Celkové trénování trvalo 9 hodin.

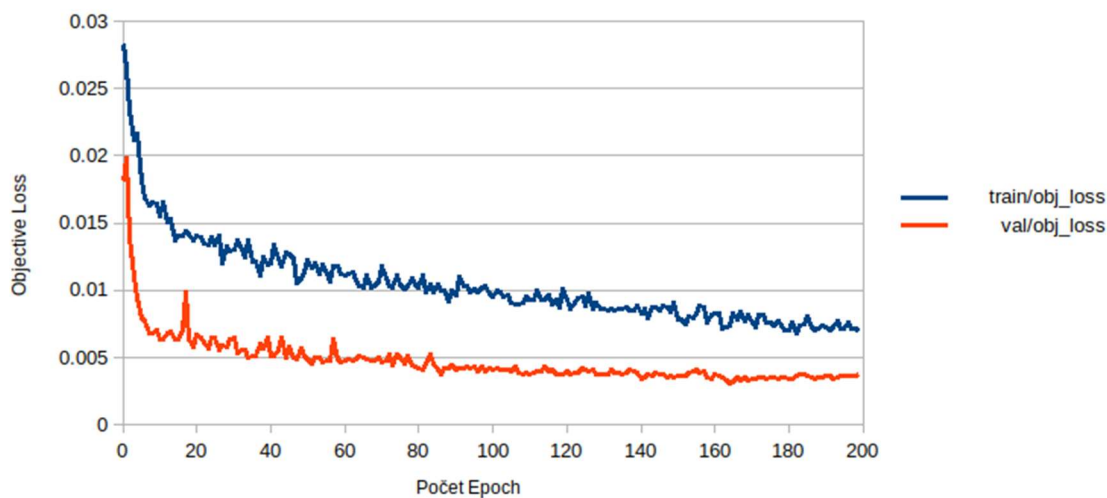
Prvek	Nejvyšší dosažená hodnota (%)
mAP_0.5	94,4
mAP_0.5:0.95	84,3
Přesnost	91,8
Úplnost	89,2

Tab. 2 - Hodnoty prvků sítě YOLOv5n

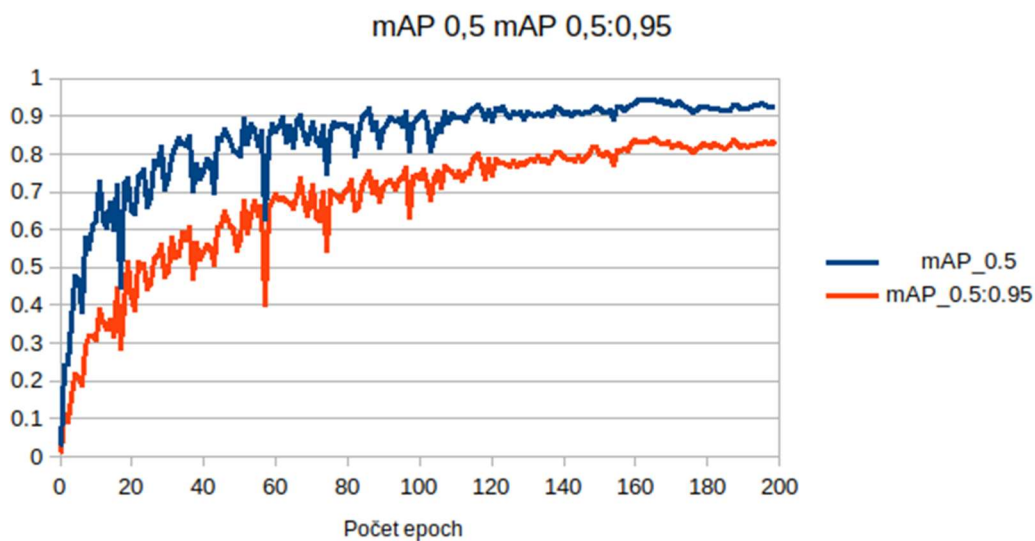
Přesnost natrénovaného modelu byla 92 %, tedy už poměrně kvalitní výsledek při použití nejmenšího datasetu. Průměrná přesnosti neboli *mAP* 50 byla 94 %.

Matice záměn						
YOLOv5n		Úplnost				
		Papír	Sklo	Plast	Kov	Pozadí
Predikce	Papír	0,86	x	0,03	x	0,17
	Sklo	x	0,78	x	x	0,17
	Plast	0,14	0,22	0,94	x	0,33
	Kov	x	x	x	1,00	0,33
	Pozadí	x	x	0,03	x	x

Tab. 3 - Matice záměn modelu YOLOv5n



Obr. 31 - Průběh ztrátové funkce na trénovacích a validačních objektech YOLOv5n



Obr. 32 - Průběh ztrátové funkce na trénovacích a validačních objektech YOLOv5n

### 10.6.2 YOLOv5s

Jedná se o malou neuronovou síť. Tato verze je menší s rychlejším tréninkem a je doporučována jako defaultní váha. Jedna epocha u tohoto modelu trvala přibližně 6 minut 30 sekund. Velikost obrázku byla zvolena 640px, *batch-size* 16 a počet epoch 200. Celkové trénování trvalo 23 hodin.

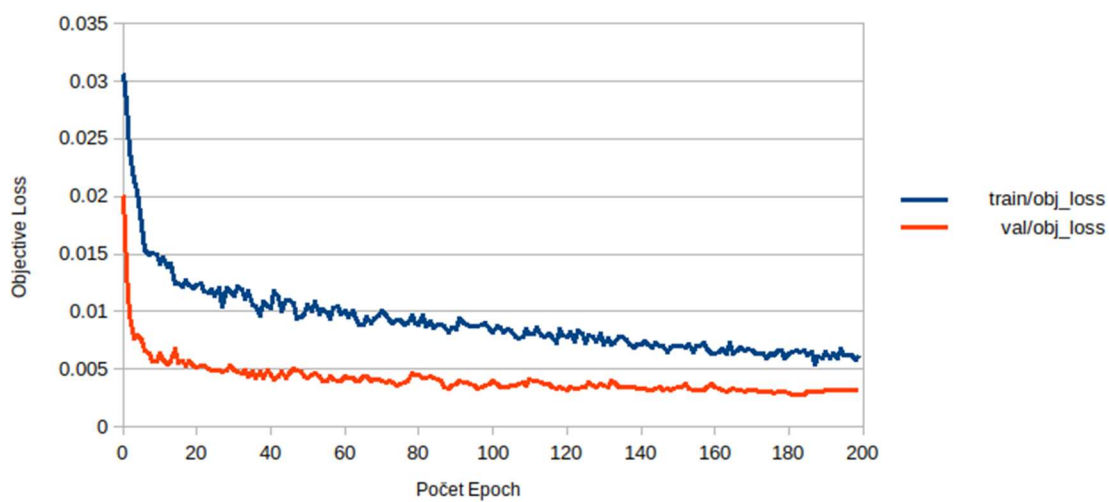
Prvek	Nejvyšší dosažená hodnota (%)
mAP_0.5	90,2
mAP_0.5:0.95	84,1
Přesnost	88,6
Úplnost	79,7

Tab. 4 - Hodnoty prvků sítě YOLOv5s

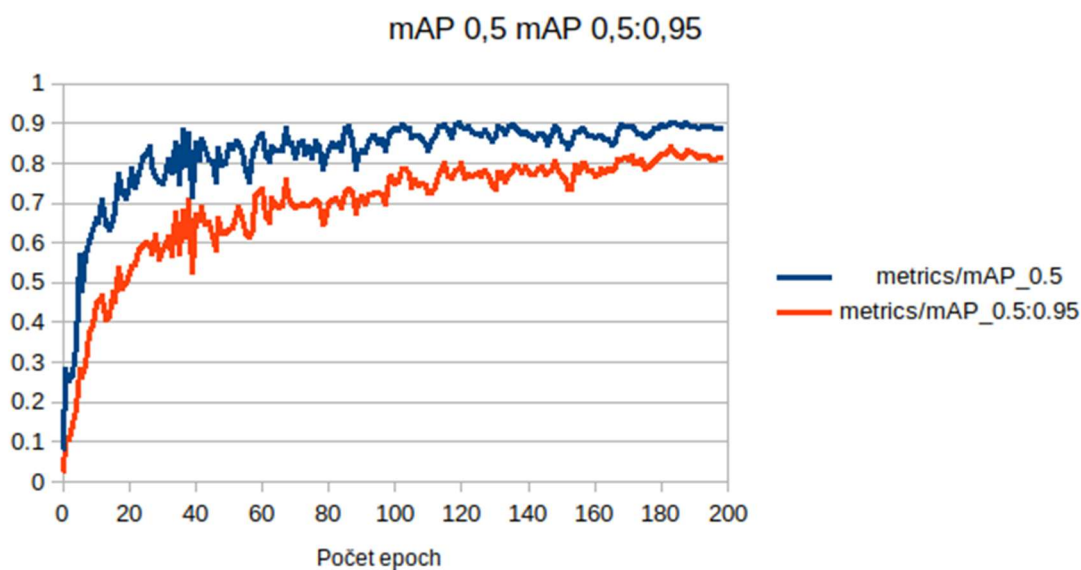
Přesnost natrénovaného modelu byla 88 % při použití menšího datasetu. Průměrná přesnost neboli mAP\_0.5 byla 90 %. Což je o něco menší než u přechozího modelu YOLOv5n.

Matice záměn						
YOLOv5s		Úplnost				
		Papír	Sklo	Plast	Kov	Pozadí
Predikce	Papír	0,95	x	x	x	0,36
	Sklo	x	0,56	x	x	0,09
	Plast	0,05	0,33	0,90	0,08	0,09
	Kov	x	0,11	0,06	0,92	0,45
	Pozadí	x	x	0,03	x	x

Tab. 5 - Matice záměn modelu YOLOv5s



Obr. 33 - Průběh ztrátové funkce na trénovacích a validačních objektech YOLOv5s.



Obr. 34 - Graf zachycující přesnost modelu YOLOv5s a chyby  $mAP_{0.5}$  a  $mAP_{0.5:0.95}$

### 10.6.3 YOLOv5m

Jedná se o střední neuronovou síť. Tato verze je střed61 napříč předtrénovanými možnostmi a měla by dosahovat lepších výsledků nežli YOLOv5s a YOLOv5n. Jedna epocha u tohoto modelu trvala přibližně 13 minut 30 sekund. Velikost obrázku byla zvolena 640px, *batch-size* 16 a počet epoch 200. Celkové trénování trvalo 60 hodin.

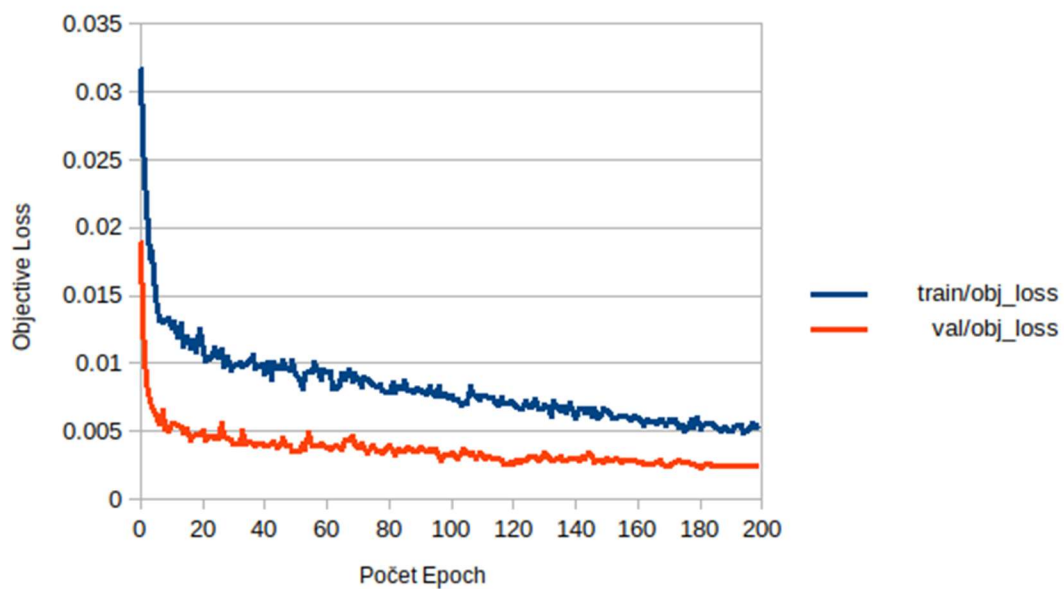
Prvek	Nejvyšší dosažená hodnota (%)
mAP_0.5	92,3
mAP_0.5:0.95	88,6
Přesnost	91,7
Úplnost	83,6

Tab. 6 - Hodnoty prvků sítě YOLOv5m

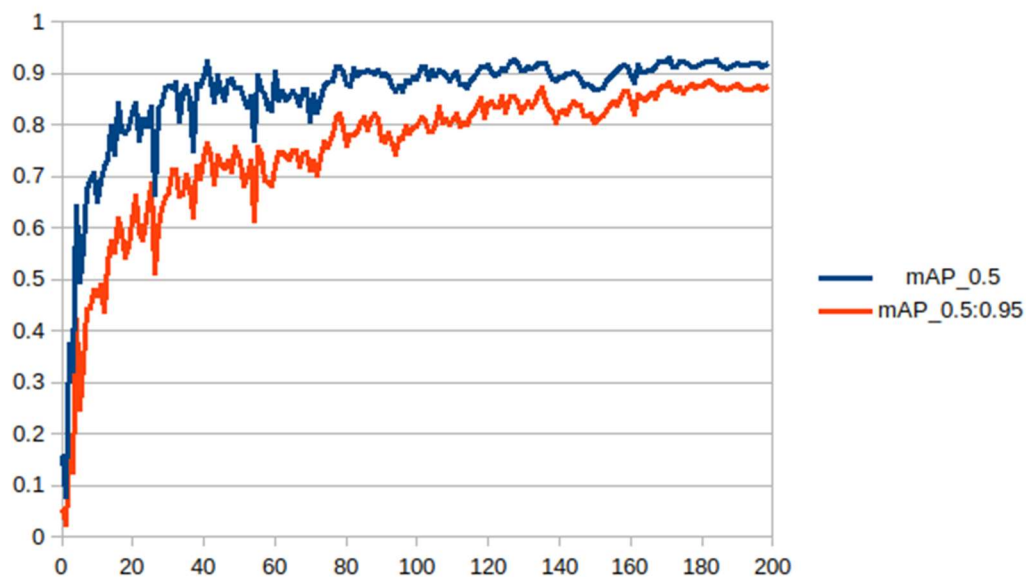
Přesnost natrénovaného modelu byla 92 %, při použití středního datasetu. Průměrná přesnost neboli  $mAP_{50}$  byla 92 %. A  $mAP_{0.5:0.95}$  vyšla 89 %, což je nejlepší výsledek ze všech modelů.

Matice záměn						
YOLOv5s		Úplnost				
		Papír	Sklo	Plast	Kov	Pozadí
Predikce	Papír	0,86	x	0,06	x	x
	Sklo	x	0,78	x	x	x
	Plast	0,09	0,22	0,90	0,17	0,50
	Kov	x	x	x	0,83	0,50
	Pozadí	0,05	x	0,03	x	x

Tab. 7 - Matice záměn modelu YOLOv5m



Obr. 35 - Průběh ztrátové funkce na trénovacích a validačních objektech YOLOv5m



Obr. 36 - Graf zachycující přesnost modelu YOLOv5m a chyby  $mAP_{0.5}$  a  $mAP_{0.5:0.95}$

#### 10.6.4 YOLOv5l a YOLOv5x

Velká a největší verze s pomalým tréninkem disponující lepší detekční přesností než předchozí modely. Pro tento model nebyly tyto modely využity z důvodu nedostatečné kapacity hardwaru, jelikož bylo přetěžované CPU a grafická karta. Následkem toho se přehříval počítač a zasekával se pod nutným restartováním celého počítače.

### 10.7 VYHODNOCENÍ DAT

Výsledné zhodnocení natrénovaných modelů s pomocí předtrénovaných modelů YOLOv5. V této kapitole dojde ke zhodnocení výsledných modelů. Všechny modely byly pouštěny s parametry velikostí snímku 640px, *batch-size* 16 a počet epoch 200.

Model	Velikost MB
YOLOv5n	3,8
YOLOv5s	14,4
YOLOv5m	42,2

Tab. 8 - Velikosti výsledných modelů

Velikost modelů roste s vyšší řadou. Čím vyšší model, tím větší je jeho výsledný objem.

Největší byl z předučeného YOLOv5m, který měl 42 MB.

Výsledná ukázka detekce na snímku, kde všechny třídy byly rozpoznány na 96 % a více. Rámečky byly zvoleny pro papír modrá, plast bílá, kov zelená a sklo tyrkysová.



Obr. 37 - Ukázka detekce

Následující sloupcový graf ukazuje tři hlavní prvky pro jednotlivé modely, a to *cls\_loss*, *box\_loss* a *obj\_loss* pro použité modely YOLOv5n, YOLOv5s a YOLOv5m, kde nejlepší výsledky má model YOLOv5m.

### ***Cls\_loss***

*Clc\_loss* neboli classification loss je ztrátová funkce, která měří, jak dobře model klasifikuje každý vyhodnocený ohraničující rámeček.

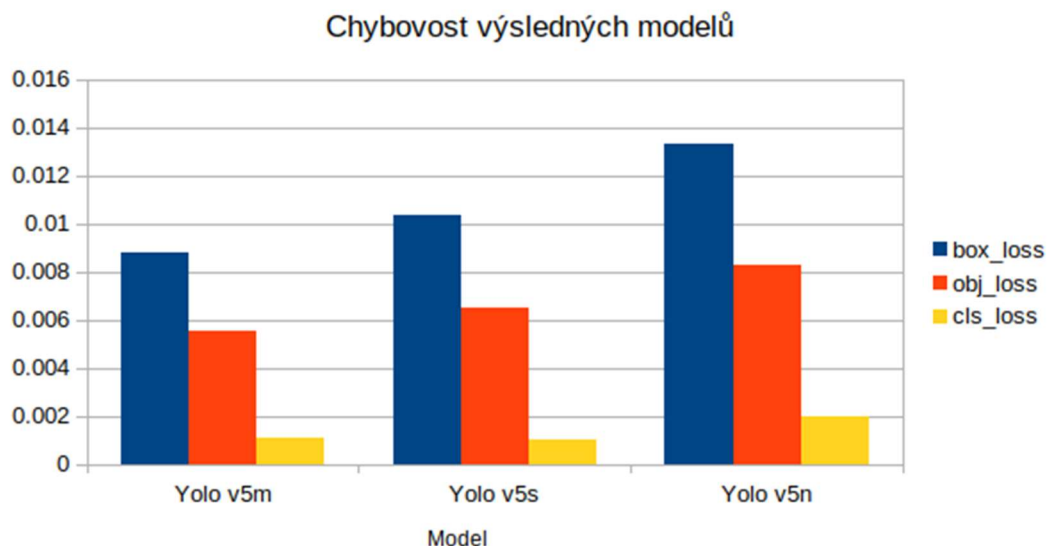
### ***Box\_loss***

Je ztrátová funkce vyhodnocující přesnost předpokládaných souřadnic vyhodnoceného rámečku vůči referenčním rámečkům.

### ***Obj\_loss***

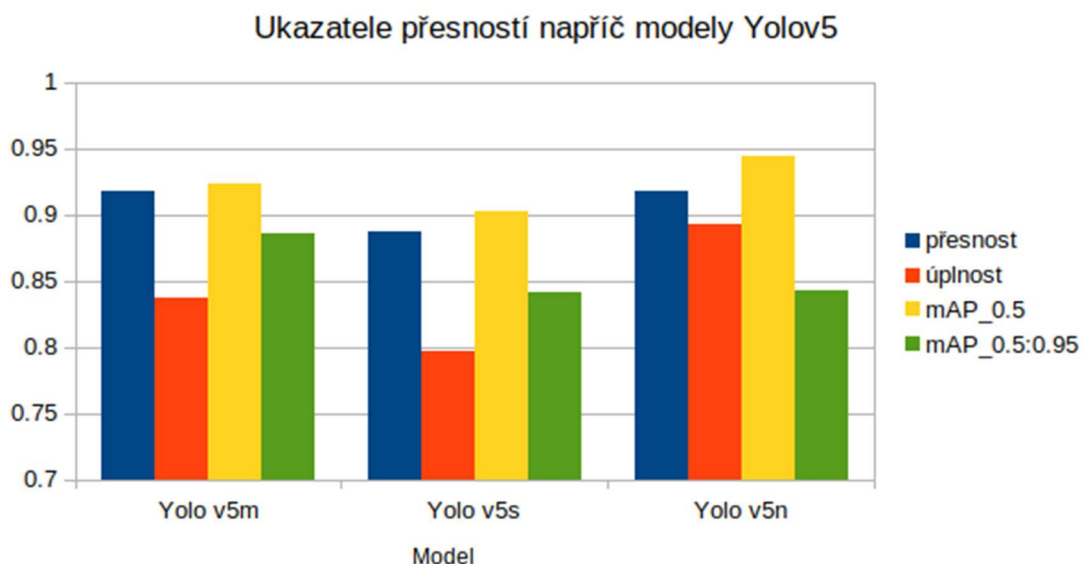
Jedná se o ztrátu objektu, která posuzuje, jak dobře model předpovídá a zda se v dané mřížce nachází objekt, či ne.

V grafu chybovosti si lze všimnout shodné chybovosti klasifikační ztráty YOLOv5m a YOLOv5s. Nejmenší chybovost má YOLOv5m.



Obr. 38 - Chybovosti výsledných modelů

Graf přesností mezi výslednými modely, kde nejvyšší přesnost a  $mAP_{0.5:0.95}$  má model z YOLOv5m a nejvyšší úplnost a  $mAP_{0.5}$  má model YOLOv5n.

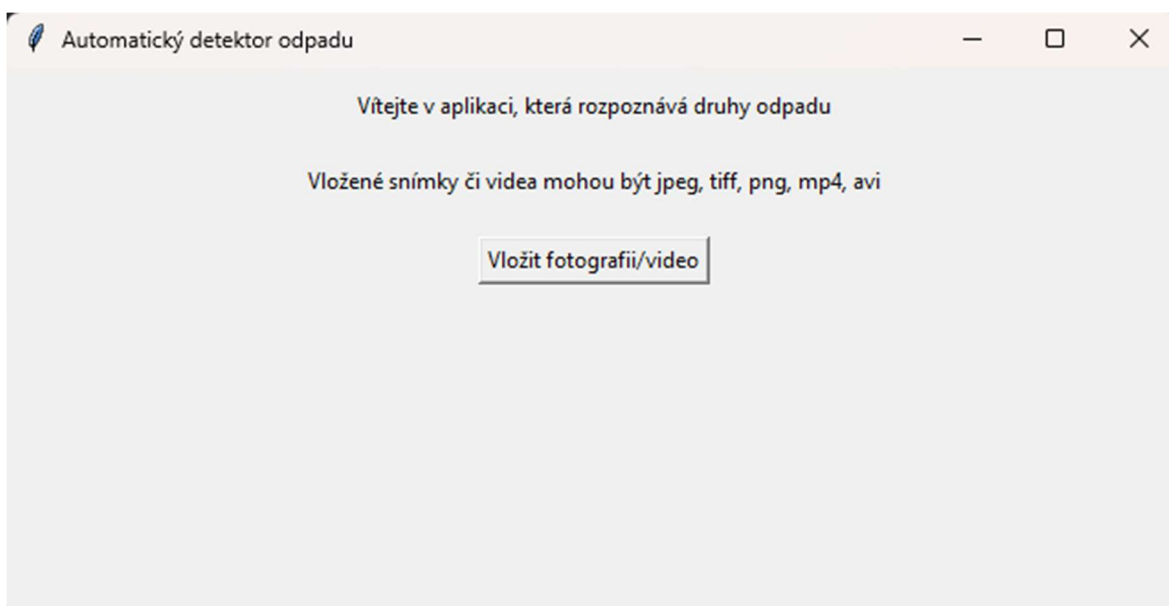


Obr. 39 - Graf přesností napříč modely

## 11 GRAFICKÉ ZPRACOVÁNÍ APLIKACE

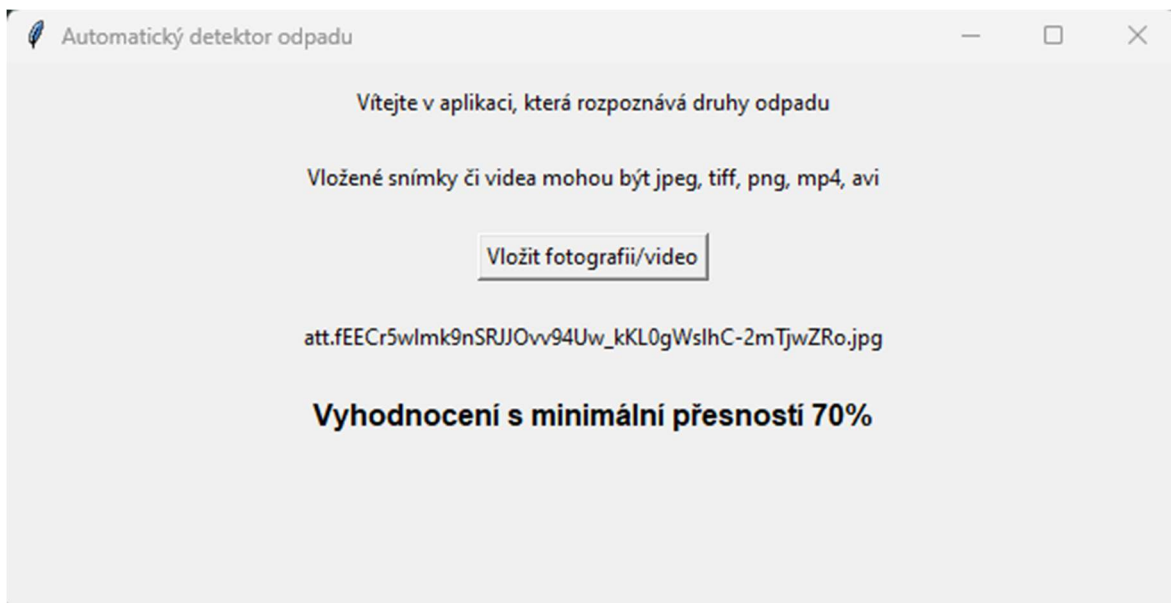
Pro lepší využití samostatné detekce byla vytvořena GUI aplikace v programovacím jazyce Python s velmi jednoduchým rozhraním.

Jako první je potřeba nahrát snímek nebo video, které musí být ve formátech .jpeg, .png, .tiff, .mp4, nebo .avi. K tomu slouží tlačítko Vložit fotografii/video. To otevře defaultní okno pro vyhledávání souboru, do kterého lze nahrát zvolený soubor.



Obr. 40 - Ukázka aplikace

Po zvolení snímku se zobrazí název vybraného souboru, s kterým se dále pracuje. Je spuštěna detekce, kde se provede skript provolávající YOLOv5 detect.py skript. Ten má parametry *weights*, což je výsledný naučený model *source*, který odkazuje na umístění souboru *conf*, což říká, s jakou minimální přesností chceme dané třídy zobrazit na snímku a poslední parametr *view-img* pro zobrazení snímku.



Obr. 41 - Ukázka aplikace po nahrání fotografie

Výsledky jsou nahrány do YOLOv5 adresáře, konkrétně do složky *runs/detect*, kde se generují po každém naučení *expXX* složky s výsledkem. *XX* je číslo generované sekvencí samotným programem YOLOv5.



Obr. 42 - Ukázka výsledné detekce plastu

## **11.1 KNIHOVNY POUŽITÉ PRO GUI**

### **11.1.1 Tkinter**

Tkinter je open-source knihovna, která se využívá při tvorbě GUI aplikací v jazyce Python. Mezi její přednosti patří jednoduchost, snadná implementace a také snadná integrace s jinými knihovnami či moduly.

Tato knihovna obsahuje sadu widgetů, tedy základní grafické prvky. Je to například Button, Label, Entry, Frame a další. Každý widget umožňuje nastavení jako je barva, šířka aj. Rozmístění widgetů poté řeší Grid, Pack nebo Place (Wikipedia, 2023).

### **11.1.2 PIL**

Python Imaging Library je o open-source knihovna pro Python, která umožňuje manipulaci s fotografiemi. Knihovna umožňuje velkou podporu pro různé druhy formátů, jejich interní reprezentaci a velké schopnosti zpracování obrazu. PIL je primárně navržen pro rychlý přístup k uloženým datům v daných pixelových formátech (Wikipedia, 2024).

## 12 ZÁVĚR

Cílem diplomové práce bylo vytvořit model pro automatickou detekci odpadu.

Teoretická část nejprve nastínila problematiku odpadu a recyklace. Dále obsahuje vybranou senzorovou techniku a lokalizaci nalezeného objektu z kamery. Také obsahuje základní principy neuronových sítí a jejich následné učení. Dále jsou popsány klasifikace objektů, jednotlivé metriky pro hodnocení modelů a popis zvoleného detekčního algoritmu.

Praktická část byla věnována nejprve popisu datasetu a celkovému shromáždění dat. Bylo zde využito celkem 489 snímků, tedy 418 snímků a 71 pro validaci. Datová sada byla tvořena v open-source programu makesense.ai. Tato sada byla použita k následnému trénování pomocí detekčního programu YOLO, konkrétně ve verzi YOLOv5 a k jejím modifikacím. Tyto jednotlivé modely byly testovány na datech a pomocí jednotlivých metrik vyhodnoceny formou tabulek a grafů.

Aplikace byla vytvořena v prostředí Python s využitím knihovny Tkinter s jednoduchým rozhraním, která umožnila snadno nahrát snímek či video a vyhodnotit jej.

Nejoptimálnější model, který byl i dále používán, byl z modifikace YOLOv5m, protože vykazoval nejmenší chybovost. Také byly vytvořeny modely z YOLOv5n a YOLOv5s, které ovšem vyšly o něco nepřesněji. Zajímavostí je shodná chybovost klasifikační ztráty u YOLOv5s a YOLOv5n.

Některé z vybraných modifikací, konkrétně YOLOv5l a YOLOv5x, nebylo možné zrealizovat z důvodů nedostatečné kapacity hardwaru.

## LITERATURA A ZDROJE

ABDULLA Sarah. *Image quality*. [online]. 2021. [cit. 2024-07-31]. Dostupné z: <https://www.radiologycafe.com/frer-physics-notes/x-ray-imaging/image-quality/>

AGARWAL Rahul. *Complete Guide to the Adam Optimization Algorithm*. [online]. 2023. [cit. 2024-07-29]. Dostupné z: <https://builtin.com/machine-learning/adam-optimization>

ARNAB, Anurag *et al.* *Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation*. [online]. 2018. [cit. 2024-07-25]. Dostupné z: <https://www.robots.ox.ac.uk/~tvlg/publications/2017/CRFMeetCNN4SemanticSegmentation.pdf>

ASHTARI, Hossein. *CCD vs. CMOS: 5 Differences To Know*. Online. In: Spiceworks. [online]. 2023. [cit. 2024-07-25]. Dostupné z: <https://www.spiceworks.com/tech/tech-general/articles/ccd-vs-cmos/>

BLAŽKOVÁ, Anežka. *OVLÁDACÍ SW ROBOTY UR3 PRO AUTOMATIZOVANÉ HRANÍ PIŠKVOREK*. Pardubice, 2020. bakalářská práce (Bc.). Univerzita Pardubice. Fakulta elektrotechniky a informatiky

BOCHKOVSKIY, Aleksey. *YOLOv4 — the most accurate real-time neural network on MS COCO dataset*. [online]. 2020. [cit. 2024-07-25]. Dostupné z: <https://alexeyab84.medium.com/yolov4-the-most-accurate-real-time-neural-network-on-ms-coco-dataset-73adfd3602fe>

BOUVRIE, Jake. *Notes on convolutional neural networks*. *Technická zpráva*, listopad 2006. [cit. 2024-07-15].

CONTROL ENGINEERING ČESKO. *Co je to strojové vidění a jak může pomoci?* [online]. 2019. [cit. 2024-07-25]. Dostupné z: <https://www.vseoprmyslu.cz/automatizace/kontrola-procesu/co-je-to-strojove-videni-a-jak-muze-pomoci.html>

CROSSVALIDATED. *What is batch size in neural network?* [online]. 2022. [cit. 2024-07-29]. Dostupné z: <https://stats.stackexchange.com/questions/153531/what-is-batch-size-in-neural-network>

DOLEJŠÍ Tomáš *Dynamický rozsah - Klíč k přepalům a podexpozičním*. [online]. 2007. [cit. 2024-07-25]. Dostupné z: <https://www.fotoradce.cz/dynamicky-rozsah-klic-k-prepalum-a-podexpozicim>

Doležel, Petr. *Úvod do umělých neuronových sítí pro studenty technických vysokých škol*. [Dokument] Pardubice: Univerzita Pardubice, 2016. ISBN 978-80-7560-022-6

EUROPEAN ENVIRONMENT AGENCY. *Waste and recycling*. [online]. 2024. [cit. 2024-07-25]. Dostupné z: <https://www.eea.europa.eu/en/topics/in-depth/waste-and-recycling?activeTab=07e50b68-8bf2-4641-ba6b-eda1afd544be>

FEEDIT. *V České republice se v minulém roce recyklovalo více obalů, Češi v třídění nepolevují*. [online]. 2024. [cit. 2024-07-25]. Dostupné z: <https://feedit.cz/2024/05/16/v-ceske-republice-se-v-minulem-roce-recyklovalo-vice-obalu-cesi-v-trideni-nepolevuji/>

FELDMAN, J. a ROJAS, R. *Neural Networks: A Systematic Introduction*. 1996. [cit. 2024-07-15].

FRÄNTI Pasi, MARIESCU-ISTODOR Radu. *Soft precision and recall*. [online]. 2023. [cit. 2024-07-25]. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0167865523000296>

GALLAGHER, James. *How to Use the Roboflow Bird Detection API*. In: Roboflow. 2023. [online]. [cit. 2024-07-22]. Dostupné z: <https://blog.roboflow.com/bird-detection-api/>

GEEKSFORGEEKS. *Introduction to Deep Learning*. [online]. 2024. [cit. 2024-07-25]. Dostupné z: <https://www.geeksforgeeks.org/introduction-deep-learning/>

GEEKSFORGEEKS. *R-CNN vs rychlé R-CNN vs rychlejší R-CNN | ML* [online]. 2023. [cit. 2024-07-25]. Dostupné z: <https://www.geeksforgeeks.org/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-ml/>

GEEKSFORGEEKS. *YOLO: You Only Look Once – Real Time Object Detection*. [online]. 2022. [cit. 2024-07-25]. Dostupné z: <https://www.geeksforgeeks.org/yolo-you-only-look-once-real-time-object-detection/>

GIBES RADIM. *Citlivost ISO: jak funguje?* [online]. 2023. [cit. 2024-07-31]. Dostupné z: <https://www.fotolab.cz/blog/citlivost-jak-to-funguje/>

GOODFELLOW, Ian; BENGIO, Yoshua and COURVILLE, Aaron. *Deep learning. Adaptive computation and machine learning*. [online] The MIT Press, Cambridge, Massachusetts, 2016. Dostupné z: <https://www.deeplearningbook.org/>.

GOPRO. Online. 2024. [cit. 2024-07-31]. Dostupné z: <https://gopro.com/en/se/shop/cameras/hero11-black/CHDHX-111-master.html>.

GREENGARD, Samuel. *What is Artificial Intelligence & How Does It Work?* [online]. 2019. [cit. 2024-07-25]. Dostupné z: <https://www.datamation.com/artificial-intelligence/what-is-artificial-intelligence/>

Institut biostatistiky a analýz Lékařské fakulty Masarykovy univerzity. *Koncept umělé neuronové sítě*. [online]. Matematická biologie. [cit. 2024-07-15]. Dostupné z: <https://portal.matematickabiologie.cz/index.php?pg=analiza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--uvod-do-neuronovych-siti--koncept-umele-neuronove-site>

Institut biostatistiky a analýz Lékařské fakulty Masarykovy univerzity. *Učení bez učitele*. [online]. Matematická biologie. [cit. 2024-07-15]. Dostupné z : <https://portal.matematickabiologie.cz/index.php?pg=analiza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--adaptacni-dynamika-neuronu--uceni-bez-ucitele>

Institut biostatistiky a analýz Lékařské fakulty Masarykovy univerzity. *Učení s učitelem*. [online]. Matematická biologie. [cit. 2024-07-15]. Dostupné z: <https://portal.matematickabiologie.cz/index.php?pg=analiza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--adaptacni-dynamika-neuronu--uceni-s-ucitelem>

JOCHER, Glenn. *YOLOv5*. [online]. Ultralytics YOLO docs, 2023. [cit. 2024-07-15]. Dostupné z: <https://docs.ultralytics.com/models/yolov5/#how-can-i-train-a-yolov5u-model-using-the-ultralytics-python-api>

KARN, Ujjwal. *An Intuitive Explanation of Convolutional Neural Networks*. [online]. the data science blog, 2016. [cit. 2024-07-15]. Dostupné z: [https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/?fbclid=IwAR3p0NfC\\_catBGii1jc4ZJnamEmsBx4tWRoIOXuy30kEQfHk5U6nMK-\\_wF0](https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/?fbclid=IwAR3p0NfC_catBGii1jc4ZJnamEmsBx4tWRoIOXuy30kEQfHk5U6nMK-_wF0)

KUBÍNEK, Jiří. *DETEKCE OBJEKTŮ V OBRAZE*. Brno. 2009 [cit. 2024-07-15].  
Dostupné také z: [https://www.vut.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=116946](https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=116946) Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Vedoucí práce Ing. Michal Hradiš.

KUMAR, Sourav. *Data Augmentation Increases Accuracy of your model — But how?* [online]. 2019. [cit. 2024-07-25]. Dostupné z: <https://medium.com/secure-and-private-ai-writing-challenge/data-augmentation-increases-accuracy-of-your-model-but-how-aa1913468722>

KUNDU Rohit. *Confusion Matrix: How To Use It & Interpret Results [Examples]*. [online]. 2022. [cit. 2024-07-25]. Dostupné z: <https://www.v7labs.com/blog/confusion-matrix-guide>

KUNDU Rohit. *YOLO: Algorithm for Object Detection Explained [+Examples]*. [online]. 2023. [cit. 2024-07-25] Dostupné z: <https://www.v7labs.com/blog/yolo-object-detection>

LANDRO NICOLA. *Mixing ADAM and SGD: a Combined Optimization Method*. [online]. 2020. [cit. 2024-07-31]. Dostupné z: <https://arxiv.org/abs/2011.08042>

LI Yixing, REN Fengbo. *Light-Weight RetinaNet for Object Detection*. [online]. 2019. [cit. 2024-07-25]. Dostupné z: <https://arxiv.org/abs/1905.10011>

MATHWORKS. *Improve Shallow Neural Network Generalization and Avoid Overfitting*, 2016. [online]. [cit. 2024-07-15]. Dostupné z: <https://www.mathworks.com/help/deeplearning/ug/improve-neural-network-generalization-and-avoid-overfitting.html;jsessionid=f7a3cf146a4a07f6fde1947af79c>

MOUJAHID, Adil. *Data Analytics and more*. [online]. 2016 [cit. 2024-07-15]. Dostupné z: <http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe>

MUREL Jacob, KAVLAKOGLU Eda. *What is a confusion matrix?* [online]. 2024. [cit. 2024-07-25]. Dostupné z: <https://www.ibm.com/topics/confusion-matrix>

NAHAVANDI Saeid, ALIZADEHSANI Roohallah, NAHAVANDI Darius, MOHAMED Shady, MOHAJER Navid, ROKONUZZAMAN Mohammad, HOSSAIN

Ibrahim. *A Comprehensive Review on Autonomous Navigation* [online]. 2022. [cit. 2024-07-25]. Dostupné z: <https://arxiv.org/pdf/2212.12808>

PAUL Vivian Amolik. *SSD Object Detection in Real Time (Deep Learning and Caffe)* [online]. 2020. [cit. 2024-07-25]. Dostupné z: <https://medium.com/acm-juit/ssd-object-detection-in-real-time-deep-learning-and-caffe-f41e40eea968>

PAWANGFG. *YOLO: You Only Look Once – Real Time Object Detection*. [online]. 2022. [cit. 2024-07-25]. Dostupné z: [https://www.geeksforgeeks.org/user/pawangfg/contributions/?itm\\_source=geeksforgeeks&itm\\_medium=article\\_author&itm\\_campaign=auth\\_user](https://www.geeksforgeeks.org/user/pawangfg/contributions/?itm_source=geeksforgeeks&itm_medium=article_author&itm_campaign=auth_user)

POTRIMBA Petru. *What is R-CNN?* [online]. 2023. [cit. 2024-07-25]. Dostupné z: <https://blog.roboflow.com/what-is-r-cnn/>

QCEXPRT. *Neuronová síť*. [online]. [cit. 2024-07-15]. Dostupné z: [https://www.trilobyte.cz/downloadfree/qcemanual/neural\\_net.pdf](https://www.trilobyte.cz/downloadfree/qcemanual/neural_net.pdf)

QUACH Da Luy, NGUYEN Quoc Khang, QUYNH Nguyen Anh, NGOC Tran Hoang. *Evaluating the Effectiveness of YOLO Models in Different Sized Object Detection and Feature-Based Classification of Small Objects*. [online]. 2023. [cit. 2024-07-25]. Dostupné z: [https://www.researchgate.net/profile/Luy-Da-Quach/publication/373899688\\_Evaluating\\_the\\_Effectiveness\\_of\\_YOLO\\_Models\\_in\\_Different\\_Sized\\_Object\\_Detection\\_and\\_Feature-Based\\_Classification\\_of\\_Small\\_Objects/links/6507cb8cca19e8355c9a7fb0/Evaluating-the-Effectiveness-of-YOLO-Models-in-Different-Sized-Object-Detection-and-Feature-Based-Classification-of-Small-Objects.pdf](https://www.researchgate.net/profile/Luy-Da-Quach/publication/373899688_Evaluating_the_Effectiveness_of_YOLO_Models_in_Different_Sized_Object_Detection_and_Feature-Based_Classification_of_Small_Objects/links/6507cb8cca19e8355c9a7fb0/Evaluating-the-Effectiveness-of-YOLO-Models-in-Different-Sized-Object-Detection-and-Feature-Based-Classification-of-Small-Objects.pdf)

RIZZOLI Alberto. *The Ultimate Guide to Object Detection*. [online]. 2021. [cit. 2024-07-25]. Dostupné z: <https://www.v7labs.com/blog/object-detection-guide>

ROSEBROCK, Adrian. *Intersection over Union (IoU) for object detection*. [online]. 2016. [cit. 2024-07-25]. Dostupné z: <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>

SAHA Sumit. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. [online]. 2018. [cit. 2024-07-31]. Dostupné z: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

SAUMYASAXENA2730. *What is Artificial Intelligence & How Does It Work?* [online]. 2024. [cit. 2024-07-25]. Dostupné z: <https://www.geeksforgeeks.org/introduction-deep-learning/>

SHAH Deval. *Intersection over Union (IoU): Definition, Calculation, Code.* [online]. 2023. [cit. 2024-07-25]. Dostupné z: <https://www.v7labs.com/blog/intersection-over-union-guide>

SHAH Deval. *Mean Average Precision (mAP) Explained: Everything You Need to Know.* [online]. 2022. [cit. 2024-07-25]. Dostupné z: <https://www.v7labs.com/blog/mean-average-precision>

SOLAWETZ Jacob. *What is Mean Average Precision (mAP) in Object Detection?* [online]. 2020. [cit. 2024-07-25]. Dostupné z: <https://blog.roboflow.com/mean-average-precision/>.

SOLAWETZ, Jacob. *What is YOLOv5? A Guide for Beginners.* [online]. 2020. [cit. 2024-07-25]. Dostupné z: <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>

SRIVASTAVA, Nitish; HINTON, Geoffrey; KRIZHEVSKY, Alex; SUTSKEVER, Ilya a SALAKHUTDINOV, Ruslan, 2014. [cit. 2024-07-15]. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research.* 15, 1929–1958

STAFF Coursera. *What Is Python Used For? A Beginner's Guide.* [online]. 2024. [cit. 2024-07-31]. Dostupné z: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>

STAFF Coursera. *What Is PyTorch?* [online]. 2024. [cit. 2024-07-31]. Dostupné z: <https://www.coursera.org/articles/what-is-pytorch>

ŠIMEČÍK JAN. *Rozlišení displeje: Co to je a jaké jsou nejčastější typy?* [online]. 2023. [cit. 2024-07-31]. Dostupné z: <https://toply.cz/pojmy/displej/rozliseni/>

TAN Mingxing, PANG Ruoming, LE Quoc V. *EfficientDet: Scalable and Efficient Object Detection.* [online]. 2020. [cit. 2024-07-25]. Dostupné z: <https://arxiv.org/abs/1911.09070>

THAKUR Ayush *What's the Optimal Batch Size to Train a Neural Network?* [online]. 2022. [cit. 2024-07-25]. Dostupné z: <https://wandb.ai/ayush-thakur/dl-question->

[bank/reports/What-s-the-Optimal-Batch-Size-to-Train-a-Neural-Network---VmlldzoyMDkyNDU](https://ultralytics.com/reports/What-s-the-Optimal-Batch-Size-to-Train-a-Neural-Network---VmlldzoyMDkyNDU)

ULTRALYTICS Inc. *Ultralytics YOLOv5 Architecture*. [online]. 2023. [cit. 2024-07-25]. Dostupné z: [https://docs.ultralytics.com/yolov5/tutorials/architecture\\_description/](https://docs.ultralytics.com/yolov5/tutorials/architecture_description/)

VOJÁČEK, Antonín. *Samoučící se neuronová síť – SOM, Kohonenovy mapy*. [online]. 2006. [cit. 2024-07-15]. Dostupné z: [https://www.kiv.zcu.cz/studies/predmety/uir/NS/Samouc\\_NN2.pdf](https://www.kiv.zcu.cz/studies/predmety/uir/NS/Samouc_NN2.pdf)

VONDRÁK, I. *Neuronové sítě*. [online] Ostrava: VŠB-TU, 1994. 56. [cit. 2024-07-15] Dostupný z: [http://vondrak.cs.vsb.cz/download/Neuronove\\_site.pdf](http://vondrak.cs.vsb.cz/download/Neuronove_site.pdf)

WALIA, Mrinal. *Semantic Segmentation vs. Instance Segmentation*. In: *Roboflow*. [online]. 2022. [cit. 2024-07-22]. Dostupné z: <https://blog.roboflow.com/difference-semantic-segmentation-instance-segmentation/>

WIKIPEDIA. *Python Imaging Library*. [online]. 2024. [cit. 2024-07-31]. Dostupné z: [https://en.wikipedia.org/wiki/Python\\_Imaging\\_Library](https://en.wikipedia.org/wiki/Python_Imaging_Library)

WIKIPEDIA. *Reprodukce (fotografie)*. [online]. 2024. [cit. 2024-07-31]. Dostupné z: [https://cs.wikipedia.org/wiki/Reprodukce\\_\(fotografie\)](https://cs.wikipedia.org/wiki/Reprodukce_(fotografie))

WIKIPEDIA. *Tkinter*. [online]. 2023. [cit. 2024-07-31]. Dostupné z: <https://cs.wikipedia.org/wiki/Tkinter>

ZHAO Nan, MURAKAMI Kenichi, YAMAKAWA Yuji. *Vision based trajectory dynamic compensation system of industrial robot*. [online]. 2024. [cit. 2024-07-25]. Dostupné z: <https://link.springer.com/article/10.1007/s00170-024-13312-5>

ZHAO Nan, MURAKAMI Kenichi, YAMAKAWA Yuji. *Vision based trajectory dynamic compensation system of industrial robot*. [online]. 2024. [cit. 2024-07-25]. Dostupné z: <https://link.springer.com/article/10.1007/s00170-024-13312-5>

## **PŘÍLOHY**

A – CD

**Příloha k diplomové práci**

Automatická detekce odpadu v obrazových datech

Bc. Anežka Blažková

CD

## **Obsah**

1. Text diplomové práce ve formátu PDF
2. Úplné zdrojové kódy pro YOLO a GUI
3. Snímky použité pro trénování sítě
4. Natrénované modely YOLOv5n, YOLOv5s, YOLOv5m