

1. Metody bez trénování

U metod bez trénování jsem vyzkoušel různé druhy filtrování obrázků (MedianBlur, Blur, GaussianBlur) v kombinaci s Sobelovým operátorem nebo Cannyho detektorem hran.

Pomocí Sobelového operátoru jsem zpočátku nedosahoval vysoké skóre jako u Cannyho, takže jsem tuto metodu dále nerozvíjel. Pomocí Cannyho detektoru hran jsem i zpočátku dosahoval vysokých hodnot a tak jsem pokračoval především touto metodou. Zkoušel jsem najít optimální řešení a optimální threshold, abych dosáhl co největšího skóre.

Filtration	Method	Score	F1 score
MedianBlur	Canny	98.7 %	97.9 %
Blur	Canny	97.6 %	96.4 %
GaussianBlur	Canny	98.2 %	97.3 %
-	Sobel	± 70 %	± 70 %

2. Metody založené na trénování

Pro prvotní trénování jsem použil knihovnu DLib, ve které jsem vyzkoušel model konvoluční sítě LeNet5.

Poté jsem pro trénování neuronových sítí jsem využil framework PyTorch. Zpočátku jsem vyzkoušel načíst předem zkonstruované modely dostupné z knihovny Torchvisions, která je obsažena ve frameworku PyTorch. Modely, které jsem vyzkoušel byly: MobileNet_V2, AlexNet, VGG11, ResNet, SqueezeNet, GoogLeNet.

Dále jsem vyzkoušel modely konvolučních sítí dostupné na PyTorch Tutorialu a jejich upravené verze z komunitních webů.

Jako poslední fázi jsem vyzkoušel zkombinovat dva nejvíce úspěšné modely v kombinaci s Cannyho detektorem hran z předchozího odstavce.

Ve všech případech jsem zkoušel nastavit různé hyper parametry jako jsou: počet epoch, batch size a velikost vstupního obrazu. Do finální tabulky jsem vybral vždy 2 nejlepší výsledky dané sítě.

Network	Epoch	Batchsize	Imgsize	Time (s)	Score	F1 score
LeNet 5	20	256	32	-	90.2 %	86.4 %
MobileNet	10	64	40	198	80.3 %	77.9 %
MobileNet	10	128	80	402	77.9 %	74.4 %
AlexNet	10	32	224	626	72.4 %	70.1 %

AlexNet	25	32	224	1975	73.4 %	71.2 %
VGG11	10	32	32	989	65.4 %	64.7 %
ResNet	10	64	40	256	72.7 %	69.4 %
ResNet	10	64	80	418	73.3 %	70.7 %
SqueezeNet	10	64	80	244	70.1 %	69.5 %
GoogLeNet	10	32	80	516	75.9 %	73.0 %
GoogLeNet	10	64	40	454	79.2 %	76.7 %
Community	10	128	80	111	96.9 %	95.2 %
Community	10	256	80	110	94.7 %	91.8 %
Community	40	256	80	427	92.4 %	88.6 %
Combined	-	-	-	-	97.9 %	96.6 %

3. Závěr

Nejlepší výsledek bez trénování jsem dosáhl kombinací Cannyho detektoru a MedianBluru (98.7 % a F1 skóre 97.9 %), na druhém místě při kombinaci dvou konvolučních sítí a Cannyho detektoru (97.9 % a F1 skóre 96.6 %). Pro metody s trénováním jsem dosáhl nejlepšího výsledku pomocí komunitní sítě (96.9 % a F1 skóre 95.2 %).

Největší problém pro metody s trénováním byly velmi tmavé obrázky a obrázky, které obsahovaly mnoho šumu. Při menším dosaženém skóre u různých sítí byl často problém ten, že falešně pozitivně označovali prázdná místa - což se u "lépe" natrénovaných sítí již nestalo. V některých případech docházelo i k špatnému rozpoznání například u vrženého stínu.

Obrázky parkovacích míst a jejich vyhodnocení Kombinovanou formou (sítě + canny) lze pozorovat na adrese: [výsledky parkoviště](#)

Níže příklady špatného rozpoznání:

