

Vysoká Škola Báňská – Technická Univerzita Ostrava
Fakulta Elektrotechniky a Informatiky
Katedra Informatiky

Analýza obrazu II

Detekce obsazenosti parkovacích míst

Obsah

1	Zadání	3
2	Detekce bez trénování	3
3	Detekce s trénováním	3
3.1	Trénovací data	3
3.2	Použité neuronové sítě	4
3.2.1	LeNet	4
3.2.2	AlexNet	4
3.3	Srovnání	4

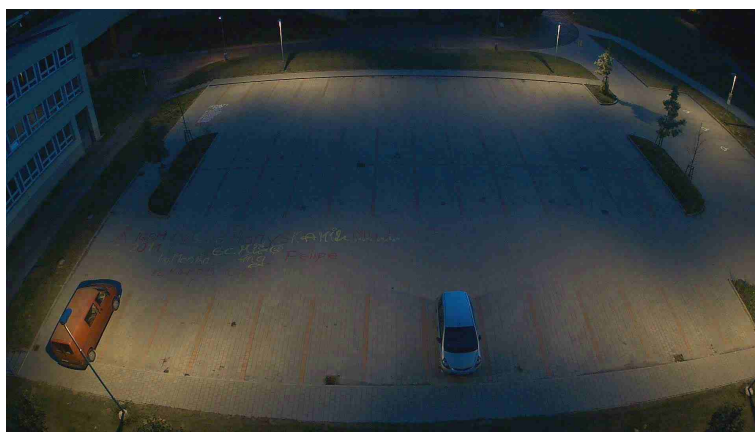
1 Zadání

Na zadaných testovacích datech detekovat obsazenost parkovacích míst, vyzkoušet metody s trénováním, bez trénování a srovnat výsledky. Pro srovnání výsledků vypsát počet nesprávně určených (false positive, false negative) výsledků a F1 score.

Testovací sada obsahuje 24 fotografií parkovacích míst, obsahující různě zaplněné parkoviště v různé denní doby.



(a) Testovací obraz



(b) Noční testovací obraz

Obrázek 1: Ukázka testovacích obrazů

2 Detekce bez trénování

3 Detekce s trénováním

3.1 Trénovací data

Pro trénování neuronových sítí byly dodány trénovací obrazy prázdných a plných parkovacích míst. Nevýhodou dodaných trénovacích dat je absence nočních snímků. Na tento nedostatek neuronové sítě trpěly, proto jsem se rozhodl implementovat umělé rozšíření datové sady. Pro výpočet rozšíření datové sady jsem využil vysoce sofistikovaného postupu, viz výpis 1. Po rozšíření trénovací sady se celkový počet trénovacích dat zosminásobil, ukázka výsledku rozšíření je zobrazen na obrázcích 2.

Po aplikaci rozšíření se zvýší doba trénování neuronové sítě, zvýší se paměťová náročnost GPU při tréninku, ale celková přesnost se podstatně zvýší (v řádu jednotek procent).

Listing 1 Výpočet rozšíření datové sady

```
void TrainInputSet::getExtendedImages(const cv::Mat &inputImg,
std::vector<cv::Mat> &extendedImages) {
    cv::Mat flippedx;
    cv::Mat flippedy;
    cv::Mat flippedxy;

    cv::flip(inputImg, flippedx, 0);
    cv::flip(inputImg, flippedy, 1);
    cv::flip(inputImg, flippedxy, -1);

    extendedImages.emplace_back(flippedx);
    extendedImages.emplace_back(flippedy);
    extendedImages.emplace_back(flippedxy);

    // Simulate night
    extendedImages.emplace_back(inputImg / 2);
    extendedImages.emplace_back(flippedx / 3);
    extendedImages.emplace_back(flippedy / 4);
    extendedImages.emplace_back(flippedxy / 5);
}
```

3.2 Použité neuronové sítě

3.2.1 LeNet

Architektura sítě

3.2.2 AlexNet

Architektura sítě

3.3 Srovnání

DNN	Velikost obrazu	Learning Rate	Min Learning Rate	Batch size	Max steps without progress	Max epochs	Doba tréninku
LeNet	28×28	0.01	10^{-6}	128	1000	300	$\approx 45s^1$
AlexNet	80×80	0.01	0.001	256	1000	300	$800s \sim 1000s$
Vgg19	32×32	0.01	10^{-7}	64	500	300	$\approx 3000s$
ResNet	32×32	0.01	10^{-7}	64	128	300	$\approx 40s$
GoogLeNet	32×32	0.01	10^{-7}	64	128	300	$\approx 35s$

Tabulka 1: Nastavení neuronových sítí

DNN	False positives	False negatives	Přesnost rozpoznání Rate	F1 score
LeNet	28×28	0.01	10^{-6}	128
AlexNet	80×80	0.01	10^{-6}	128
Vgg19	32×32	0.01	10^{-7}	128
ResNet	32×32	0.01	10^{-7}	128

Tabulka 2: Výsledky neuronových sítí

Přílohy

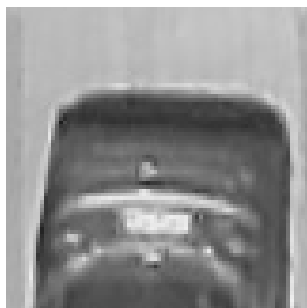
Srovnání neuronových sítí

Rozšířená trénovací sada

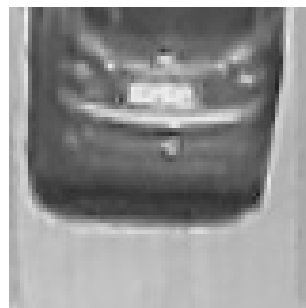
¹Trénování a rozpoznávání bylo testováno na Manjaro Linux x86_64 5.8.18, Intel i5-5200U@2.7GHz, NVIDIA GeForce GTX 850M, 4GB VRAM



(a) Originál



(b) Překlopený obraz podle x



(c) Překlopený obraz podle y



(d) Překlopený obraz podle xy



(e) Noční obraz s jase $m \frac{1}{2}$



(f) Noční obraz s jase $m \frac{1}{3}$



(g) Noční obraz s jase $m \frac{1}{4}$



(h) Noční obraz s jase $m \frac{1}{5}$

Obrázek 2: Ukázka rozšířené trénovací sady