|  |  |  |
| --- | --- | --- |
|  | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Rozpoznání SPZ** | | |
| Matěj Říčný | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2020/2021 | |

***Poděkování***

*Děkuji panu učiteli Ing. Petru Grussmannovi za cenné rady.*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2020

*podpis autora práce*

**ANOTACE**

Projekt se zabývá rozpoznáním SPZ se zobrazením dat na webové stránce. Program je napsaný v jazyce Python. Základem je OpenCV (Open Source Computer Vision Library), který zajišťuje především počítačové vidění a zpracování obrazu v reálném čase, dále model YOLOv4, který zajišťuje detekci objektů. Program funguje na principu, že YOLOv4 detektuje SPZ, poté se spustí několik technik pro zlehčení čtení znaků z obrázku, jako například zvýšení stupně šedi. Následně může tesseract lehce přečíst z obrázku SPZ, konvertovat obrázek do textu a následně uložit do MySQL databáze a dále je přečtená SPZ poslána pomocí SMTP serveru na uživatelský email. Všechna data jsou zobrazena na webové stránce napsané v aplikačním frameworku Django.

OBSAH

[Úvod 5](#_Toc61253234)

[1 Vývoj aplikace 6](#_Toc61253235)

[2 Využité technologie 7](#_Toc61253236)

[2.1 Software 7](#_Toc61253237)

[2.1.1 OpenCV 7](#_Toc61253238)

[2.1.2 YOLOv4 7](#_Toc61253239)

[2.1.3 Tesseract OCR 7](#_Toc61253240)

[2.1.4 Python 8](#_Toc61253241)

[2.1.5 XAMPP 8](#_Toc61253242)

[2.1.6 Django 8](#_Toc61253243)

[2.1.7 TensorFlow 8](#_Toc61253244)

[2.2 HARDWARE 8](#_Toc61253245)

[3 Způsoby řešení a použité postupy 9](#_Toc61253246)

[3.1 Program pro rozpoznání SPZ 9](#_Toc61253247)

[3.2 Spouštění programu 12](#_Toc61253248)

[3.3 Databáze 12](#_Toc61253249)

[4 Výsledky řešení, výstupy 13](#_Toc61253250)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 15](#_Toc61253251)

[Seznam příloh 16](#_Toc61253252)

Úvod

Jako závěrečný projekt jsem si chtěl vybrat bezpečnostní systém, ale dlouho jsem se rozhodoval, který bezpečností systém si vybrat, a nakonec jsem se rozhodl zprovoznit program, který bude rozpoznávat SPZ, protože jsem chtěl vyzkoušet pracovat s technologií openCV a detekci objektů YoloV4, protože mi to přišlo velmi zajímavé a chtěl jsem si to sám vyzkoušet. Program poté co rozpozná SPZ, upozorní uživatele e-mailem a uloží danou SPZ do databáze.

Hlavním cílem projektu bylo propojit program s databází a webovou aplikací, kde bude moct uživatel sledovat kdy byla daná SPZ rozpoznána a kdo je jejím vlastníkem, popřípadě neznámý. Uživatel je oprávněn přidávat nové vlastníky aut do databáze ve webové aplikaci a taky je mazat.

V této dokumentaci se snažím vysvětlit, jak program pro rozpoznání detekované SPZ funguje. V dokumentaci zmiňuji použité technologie, co je uživateli možno sledovat na webové stránce a v úplném závěru zmiňuji možná vylepšení.

# Vývoj aplikace

Prvním krokem bylo zprovoznit program napsaný v jazyce Python pro detekci SPZ. Poté jsem se musel naučit a zkoumat, jak program funguje, abych mohl pracovat s detekovanou SPZ.

Po zprovoznění programu jsem musel zařídit zdroj pro detekování. Jako zdroj jsem použil mobilní telefon s aplikací IP Webcam, která mi zařídila stream videa v síti WiFi.

Po implementování streamu s programem, následovala práce s daty. Program po rozpoznání SPZ pošle data obsahující jméno, příjmení, SPZ, čas a datum do MySQL databáze.

Když byla data úspěšně uložena, dalším úkolem bylo data zobrazit na webovské aplikaci. Web je napsaný v aplikačním frameworku Django, kvůli lehké implementace s daty MySQL.

Ve webovské aplikaci jsem se rozhodl udělat 3 stránky:

* První stránka webovské aplikace zobrazuje data všech zaznamenaných SPZ, čas a datum, kdy byla zaznamenána a majitele auta v tabulce.
* Druhá stránka obsahuje formulář pro vytvoření nových vlastníků aut, který obsahuje jméno, příjmení vlastníka a SPZ jeho auta.
* Třetí stránka obsahuje jméno a příjmení vlastníka auta s danou SPZ. Uživateli je umožněno vlastníka z databáze smazat.

Kromě dat na webové stránce jsou i data posílána na uživatelský email pokaždé, kdy je SPZ detekována. Email je poslán v jazyce Python, pomocí SMTP serveru a protokolu SSL.

# Využité technologie

## Software

### OpenCV

Pomocí open source knihovny OpenCV bylo umožněno zrychlení používání strojového vnímání. OpenCV. Knihovna má více než 2 500 optimalizovaných algoritmů, které zahrnují ucelenou sadu klasických i nejmodernějších algoritmů počítačového vidění a strojového učení. Tyto algoritmy lze použít k detekci a rozpoznávání tváří, identifikaci objektů, klasifikaci lidské činnosti ve videích, sledování pohybů kamer, díky kterého bylo možné detekovat SPZ. Aby byl schopen Tesseract OCR SPZ přečíst, musela knihovna OpenCV zajistit zpracování videa a obrázku.

### YOLOv4

Algoritmus YOLOv4 je založen na CNN (Convolutional Neural Network). Zajišťuje detekci a rozpoznávání objektů v reálném čase. Systém rozpoznávání v reálném čase rozpozná z obrázku více objektů a také vytvoří ohraničující rámeček kolem objektu. Yolo je založeno na jediné neurální síti (CNN). CNN rozdělí obrázek na oblasti a poté předpovídá hraniční pole a pravděpodobnosti pro každou oblast. Současně předpovídá více ohraničujících rámečků a pravděpodobností pro tyto třídy. V programu model YOLOv4 používá TensorFlow.

### Tesseract OCR

Pro čtení SPZ byl nezbytný software Tesseract OCR (Optical Character Recognition), který dokáže rozpoznávat znaky z obrázku. Proto, aby Tesseract OCR dobře konvertoval SPZ na text, musí být v dobré kvalitě, zvětšený a musí být opravena jakákoli rotace nebo zkosení.

### Python

Program pro umožnění čtení SPZ je napsaný v jazyce Python verze 3.6. Pro ukládání dat do databáze MySQL je použit mysqlconnector, který umožnil komunikaci mezi programovacím jazykem Python a databází MySQL. Řešení posílání dat emailem je řešeno pomocí SMTP serveru a protokolu SSL.

### XAMPP

Pro spuštění databáze na serveru jsem použil multiplatformní softwarový balíček XAMPP, díky kterému bylo možné vytvořit lokální server. Balíček obsahuje serverovou aplikaci Apache a databázi MySQL, do které jsem data ukládal.

### Django

Pro vytvoření webové stránky jsem se rozhodl použít webový aplikační framework Django. Django jsem se rozhodl použít z důvodu lehkého a rychlého vývoje webové stránky a jednoduchosti získávání dat z MySQL databáze.

### TensorFlow

TensorFlow je softwarová knihovna určená pro strojové učení, má vysoký výkon a běží na více procesorech a GPU.

## HARDWARE

Jako nahrávací zařízení pro detekci jsem prozatím použil mobilní telefon, do kterého jsem stáhl aplikaci IP Webcam, která mi umožnila bez problémů stream videa v síti WiFi.

# Způsoby řešení a použité postupy

## Program pro rozpoznání SPZ

##### První krok rozpoznání je převzetí souřadnic YOLOv4 pro oříznutí SPZ a přidání 5 pixelů na každé straně.

box = img[int(ymin)-5:int(ymax)+5, int(xmin)-5:int(xmax)+5]

##### Pro zlepšení čtení obrázku zvětšíme SPZ o trojnásobek pomocí funkce *cv2.resize().*

gray = cv2.resize(gray, None, fx = 3, fy = 3, interpolation = cv2.INTER\_CUBIC)

##### Přidáme obrázku stupně šedi a použijeme pro vyhlazení Gaussovo rozostření.

gray = cv2.cvtColor(box, cv2.COLOR\_RGB2GRAY)

blur = cv2.GaussianBlur(gray, (5,5), 0)



##### Pro rozlišení pozadí a popředí použijeme metodu *Otsu*, znaky SPZ jsou rozlišovány bílou barvou a pozadí černou.

ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH\_OTSU | cv2.THRESH\_BINARY\_INV)



##### Další metodou vytvoříme znaky SPZ širší pro zlepšení hledání znaků.

rect\_kern = cv2.getStructuringElement(cv2.MORPH\_RECT, (5,5))

dilation = cv2.dilate(thresh, rect\_kern, iterations = 1)

##### V dalším kroku použijeme OpenCV, aby našlo všechny znaky SPZ a ohraničilo je a seřadilo je z leva doprava.

contours, hierarchy = cv2.findContours(dilation, cv2.RETR\_T REE, cv2.CHAIN\_APPROX\_SIMPLE

sorted\_contours = sorted(contours, key=**lambda** ctr:

cv2.boundingRect(ctr)[0])



##### Ale kromě ohraničených znaků SPZ je ohraničený i znak státu, který ale nepotřebujeme, proto jej v dalším kroku vyfiltrujeme. Abychom znak odfiltrovali použijeme několik parametrů, které je třeba splnit. Tyto parametry jsou poměr výšky a šířky.

for cnt in sorted\_contours:

        x,y,w,h = cv2.boundingRect(cnt)

        height, width = im2.shape

        if height / float(h) > 6: continue

        ratio = h / float(w)

        if ratio < 1.5: continue

        if width / float(w) > 15: continue

        area = h \* w

        if area < 100: continue



##### Nyní už jsou vybrány pouze jednotlivé znaky SPZ. Oddělíme každý znak a použijeme masku *bitwise\_not* pro překlopení obrázku na černý text na bílém pozadí, aby byl Tesseract co nejpřesnější.

roi = thresh[y-5:y+h+5, x-5:x+w+5]

roi = cv2.bitwise\_not(roi)



##### Posledním krokem před posíláním obrázku pro Tesseract je použití *medianBlur* pro rozostření obrázku.

roi = cv2.medianBlur(roi, 5)

##### Poté je obrázek předán Tesseractu, aby z něj získal jednotlivé znaky a na závěr už jsou jenom jednotlivé znaky spojeny do řetězce.

text = pytesseract.image\_to\_string(roi, config='-c tessedit\_char\_whitelist=0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ --psm 8 --oem 3')

clean\_text = re.sub('[\W\_]+', '', text)

plate\_num += clean\_text

## Spouštění programu

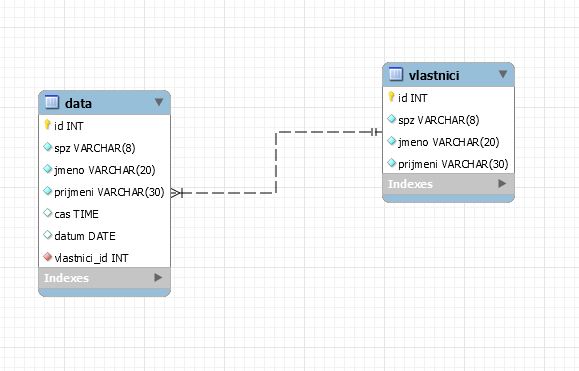
python detect\_video.py –weights ./checkpoints/custom-416 --size 416 --model yolov4 --video http://192.168.1.134:8080/video --output ./detections/recognition.avi --plate

* Weights: načtení modelu yolov4.
* Size: změnit velikost obrázku.
* Model: použití modelu yolov4.
* Video: adresář kde se video nachází, adresa streamu, nebo použití 0 pro webkameru.
* Output: kde se se rozpoznané video uloží.
* Plate: pro spuštění programu pro detekování SPZ.

## Databáze

Pro celou aplikaci jsem použil databázi MySQL, do které pomocí mysql.connector ukládám data o rozpoznaných SPZ, které obsahují SPZ, jméno vlastníka auta, příjmení vlastníka auta, čas a datum kdy byla rozpoznána. Jméno a příjmení vlastníka auta je zjištěno pomocí pomocné tabulky, kde jsou uloženy údaje o vlastnicích aut s danou SPZ.

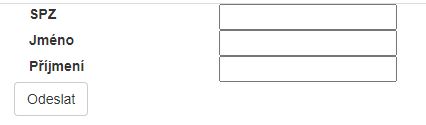
Na webu jsou data z databáze zobrazeny pomocí datových modelů z frameworku Django.



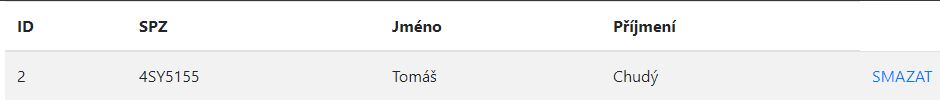
*Obrázek č.1 Databáze*

# Výsledky řešení, výstupy

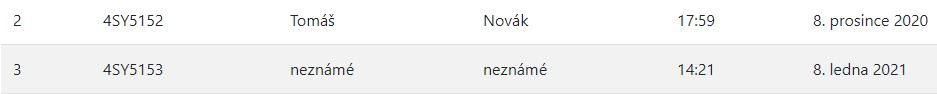
Systém je v současné době zcela funkční. Program rozpozná ze streamu SPZ a uloží do databáze. Na webové stránce jsou zobrazena data všech SPZ získána z databáze, které byly programem zaznamenány. Každá zaznamenaná SPZ je poslána emailem uživateli. Uživatel, může na webové stránce přidávat vlastníky aut do databáze, nebo je smazat.



*Obrázek č.2 Formulář*



*Obrázek č.3 Vlastníci aut*



*Obrázek č.4 Data o SPZ*

**Závěr**

Úkolem bylo zprovoznit program pro rozpoznání SPZ a její následné uložení do MySQL databáze. Program rozezná SPZ a následně ji úspěšně pošle na email, a ukládá data na MySQL databázi.

S projektem jsem spokojen, i když mám v plánu hned několik vylepšení. Mám v plánu spustit aplikaci na Colaboratory, což je produkt společnosti Google. Colaboratory mi umožní spustit kód na cloud a poskytuje bezplatný přístup k výpočetním zdrojům včetně GPU. Dále plánuji vyměnit nahrávací zařízení, místo mobilního telefonu použít IP kameru.

https://github.com/it1720/zaverecny\_projekt

Seznam použitýCH INFORMAČNÍCH ZDROJů

[1] Program pro rozpoznávání SPZ [online]. [cit. 2021-01-09].

Dostupné z: https://github.com/theAIGuysCode/yolov4-custom-functions

[2] Vložení dat do MySQL Python [online]. [cit. 2021-01-09]. Dostupné z: https://www.w3schools.com/python/python\_mysql\_insert.asp

[3] Python email [online]. [cit. 2021-01-09]. Dostupné z: https://blog.mailtrap.io/sending-emails-in-python-tutorial-with-code-examples/

[4] Rozpoznání SPZ [online]. [cit. 2021-01-09]. Dostupné z: <https://www.youtube.com/watch?v=nOIVxi5yurE&ab_channel=TheAIGuy>

[5] Rozpoznání SPZ (2) [online]. [cit. 2021-01-09]. Dostupné z: <https://www.youtube.com/watch?v=AAPZLK41rek&t=662s&ab_channel=TheAIGuy>

[6] YoloV4 [online]. [cit. 2021-01-09]. Dostupné z: <https://towardsdatascience.com/whats-new-in-yolov4-323364bb3ad3>

[7] OpenCV [online]. [cit. 2021-01-9]. Dostupné z: https://opencv.org/about/