Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Diplomová práce

Anotace obrazových dat pomocí nástroje CAPTCHA

Bc. Jan Bláha

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Jan Bláha

Informatika

Název práce

Anotace obrazových dat pomocí nástroje CAPTCHA

Název anglicky

Image data anotation using the CAPTCHA tool

Cíle práce

Cílem této diplomové práce je návrh a vytvoření aplikace, která využívá odpovědi uživatelů na ověření pomocí nástroje CAPTCHA k anotaci obrazových dat.
Dílčím cílem je automatizovaná úprava obrazových dat do vhodného formátu pro použití v CAPTCHA aplikaci.
Dalším dílčím cílem je otestování aplikace na vybraných obrazových datech.

Metodika

Nejprve bude provedena analýza stávajících řešení v oblasti ověření uživatele s důrazem na technologii CAPTCHA společně s rešerší ohledně problematiky anotace dat. Na základě analýzy bude navržena a naprogramována aplikace CAPTCHA využívající odpovědi od uživatelů k anotaci obrazových dat. Aplikace bude otestována na satelitních snímcích vybrané vesnice. Pomocí aplikace budou anotována data týkající se výskytu solárních panelů na rodinných domech. Posledním krokem bude vyhodnocení funkcionality a návrh dalšího možného rozvoje.

Doporučený rozsah práce

60-80

Klíčová slova

CAPTCHA, anotace, obrazová data, python, labelování dat, satelitní snímky, ověření uživatele

---

Doporučené zdroje informací

BRODIĆ, Darko a AMELIO, Alessia. The CAPTCHA: Perspectives and Challenges. 2020. Springer Cham, 2020. ISBN 978-3-030-29347-5.

CANTY, Morton John. *Image analysis, classification and change detection in remote sensing : with algorithms for Python.* Boca Raton ; London ; New York: CRC Press, Taylor & Francis Group, 2019. ISBN 978-1-138-61322-5.

DEY, Sandipan. Hands-on image processing with Python: expert techniques for advanced image analysis and effective interpretation of image data. Birmingham: Packt Publishing, Limited, 2018. ISBN 978-178-9343-731.

CHEN, Daniel Y. *Pandas for everyone : Python data analysis.* Boston: Addison-Wesley, 2018. ISBN 9780134546933.

PECINOVSKÝ, Rudolf. *Začínáme programovat v jazyku Python.* Praha: Grada Publishing, 2022. ISBN 978-80-271-3609-4.

Předběžný termín obhajoby

2023/24 LS – PEF

Vedoucí práce

Ing. Jakub Konopásek, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 30. 3. 2024

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 31. 3. 2024

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 31. 03. 2024

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Anotace obrazových dat pomocí nástroje CAPTCHA" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2024 _____

Poděkování

Rád bych touto cestou poděkoval vedoucímu práce Ing. Jakubu Konopáskovi, že se vedení mé práce ujal a byl mi nápomocný. Dále bych chtěl poděkovat mé snoubence, Michaele Zabilkové, která mě podporovala po dobu psaní práce. V poslední řadě bych rád poděkoval svým kolegům, kteří mi posloužili jako testovací subjekty pro otestování aplikace.

Anotace obrazových dat pomocí nástroje CAPTCHA

Abstrakt

V této diplomové práci je řešena problematika ověření uživatelů, přesněji metodu CAPTCHA. Nejprve je provedena analýza stávajících řešení práce v oblasti ověření uživatele s důrazem na metodu CAPTCHA. Na základě analýzy je navržena a naprogramována aplikace CAPTCHA využívající odpovědi uživatel k anotaci předložených obrázkových dat. Aplikace je otestována na satelitních snímcích malé vesnice. Anotovány jsou výskyty solárních panelů na domech. Na základě testů jsou vydefinovány nedostatky aplikace a doporučené úpravy.

Klíčová slova: CAPTCHA, anotace, obrazová data, python, labelování dat, satelitní snímky, ověření uživatele

Image data anotation using the CAPTCHA tool

Abstract

This diploma thesis deals with the issue of user authentication, more precisely the CAPTCHA method. First, an analysis of existing work solutions in the field of user authentication is performed, with an emphasis on the CAPTCHA method. Based on the analysis, a CAPTCHA application is designed and programmed using the user's answers to annotate the submitted image data. The application is tested on satellite images of a small village. Occurrences of solar panels on houses are annotated. Based on the tests, application deficiencies and recommended modifications are defined.

Keywords: CAPTCHA, annotation, image data, python, data labelling, satellite images, user authentication

Obsah

1 Úvod

V dnešní digitální éře, kde internet hraje klíčovou roli v našich životech a skrze něj jsou sdíleny citlivé osobní informace a údaje, je bezpečnost online prostředí důležitým oborem počítačových věd. Není překvapivé, že se k přihlašovacím údajům snaží pomocí různých metod nepovolané osoby dostat, a vyvíjí automatizované systémy, které se o zjištění údajů pokouší. Přihlašovací stránky už se proto neskládají pouze z přihlašovacího jména a hesla, ale přibývají metody k zajištění vyšší bezpečnosti. Jednou z tradičních metod zajišťující vyšší ochranu přístupových údajů je CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). Jak vyplývá z anglického názvu, cílem je oddělit lidské uživatele od automatizovaných botů pomocí testů, které jsou obtížné pro počítače, ale snadné pro lidi.

Nicméně, s vývojem technologií se schopnosti botů zlepšují a je stále obtížnější upravovat systém CAPTCHA tak, aby je odhalil. Vznikají nové typy testů, avšak vždy je jen otázkou času, než je možné je automatizovaně řešit.

Existují ale také otázky a úkoly, které by bylo možné a vhodné automaticky řešit, avšak není pro tvorbu automatizovaných systémů dostatek vhodných dat, na kterých by byly učeny. Data pro učení modelů je třeba připravit, anotovat, což je práce, která vyžaduje lidského uživatele. Nabízí se využití času lidských uživatelů řešících testy předkládané při CAPTCHA k přípravě dat pro potřeby učení modelů.

Teoretická práce se zabývá aktuálními metodami ověření uživatelů a CAPTCHA. Na základě těchto znalostí je následně navržen a naprogramován systém CAPTCHA, který využívá odpovědi od uživatelů k anotaci obrazových dat. Při řešení testu pak není lidská práce pouze prostředkem k rozlišení mezi lidmi a boty, ale také přispívá k vylepšení automatizovaných systémů.(1)(2)(3)

2 Cíl práce a metodika

2.1 Cíl práce

Cílem této diplomové práce je návrh a vytvoření aplikace, která využívá odpovědi uživatelů na ověření pomocí nástroje CAPTCHA k anotaci obrazových dat. Dílčím cílem je automatizovaná úprava obrazových dat do vhodného formátu pro použití v CAPTCHA aplikaci. Dalším dílčím cílem je otestování aplikace na vybraných obrazových datech.

2.2 Metodika

Nejprve bude provedena analýza stávajících řešení v oblasti ověření uživatele s důrazem na technologii CAPTCHA společně s rešerší ohledně problematiky anotace dat. Na základě analýzy bude navržena a naprogramována aplikace CAPTCHA využívající odpovědi od uživatelů k anotaci obrazových dat. Aplikace bude otestována na satelitních snímcích vybrané vesnice. Pomocí aplikace budou anotována data týkající se výskytu solárních panelů na rodinných domech. Posledním krokem bude vyhodnocení funkcionality a návrh dalšího možného rozvoje.

3 Teoretická východiska

S postupných rozvojem počítačů se do digitálního světa začaly přesouvat všechny informace. To, co bylo kdysi napsáno na papíře a uloženo v bezpečí domova, se dnes nachází na datových nosičích a v cloudu. K informacím je umožněn přístup téměř odkudkoliv. Je možné je prohlížet, upravovat, sdílet z mobilního zařízení v metru, z osobního počítače na dovolené, nebo ze stolního počítače v pohodlí domova. Vlastníci těchto dat však nejsou jediní, kteří o ně mají zájem. Jak se množství důležitých citlivých údajů přesouvá na internet, roste nebezpečí kybernetických útoků. Osobní informace jsou významnou komoditou a hackeři se je snaží získat, aby je mohli prodat či zneužít.

Počet kybernetických incidentů v posledních letech rapidně roste. V Roce 2023 evidoval Národní úřad pro kybernetickou a informační bezpečnost (NÚKIB) rekordní počet 262 kybernetických incidentů což je v porovnání s rokem 2022 téměř dvojnásobek.

Aby nebylo snadné se k těmto datům dostat, existují metody sloužící k ověření uživatele. Ty mají za cíl zajistit, že se k informacím dostane pouze a jen oprávněná osoba. Neoprávněné osob se nevzdávají a vymýšlí nové cesty, jak zabezpečení obejít a k datům se dostat. Je tedy třeba na zabezpečení neustále pracovat a zajišťovat nové metody, které jim jejich snahy znepříjemní.(4)(5)

3.1 Ověření uživatele

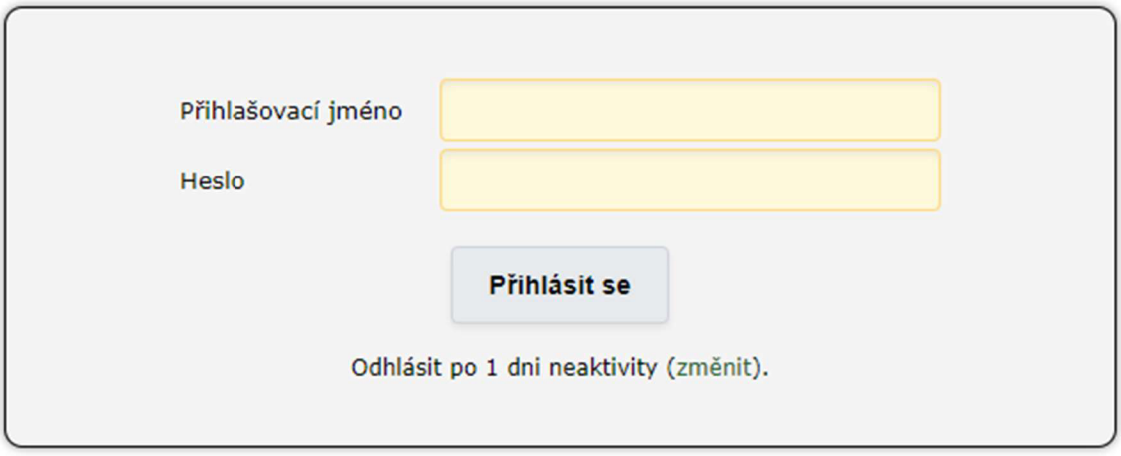
Aby bylo zajištěno, že se k datům dostanou pouze oprávněné osoby, vznikly metody pro ověření uživatele.

3.1.1 Autentizace založená na hesle

Nejjednodušší a nejvíce obvyklou metodou ověření uživatele je autentizace založená na hesle. Jak z názvu vyplívá, požaduje po oprávněném uživateli znalost hesla. Obvykle se jedná o kombinaci uživatelského jména a hesla, avšak může být i v podobě pinu, nebo přístupového kódu.

Z hlediska bezpečnosti se jedná o nejslabší ověřovací metodu. Častou chybou je nevhodně zvolené heslo, které je obvyklé a jednoduché, případně užívání stejného hesla na více místech. Další častou chybou je poznamenání hesla na nezabezpečené místo, či jeho předání jiné osobě přes nezabezpečený kanál. Při implementaci je proto důležité zvolit správná omezení pro zvolené heslo – délka, počet speciálních znaků, eliminace obvyklých frází, požadavek na změnu hesla za určité období.

Samotná autentizace založená na hesle může být obohacena dalšími metodami a funkcionalitami, které její bezpečnost zvýší. Je možné sledovat rychlost, jakou uživatel požadavky pro přihlášení odesílá, nebo jak často a kolikrát po sobě špatné heslo zadal.(2)(6)(3)

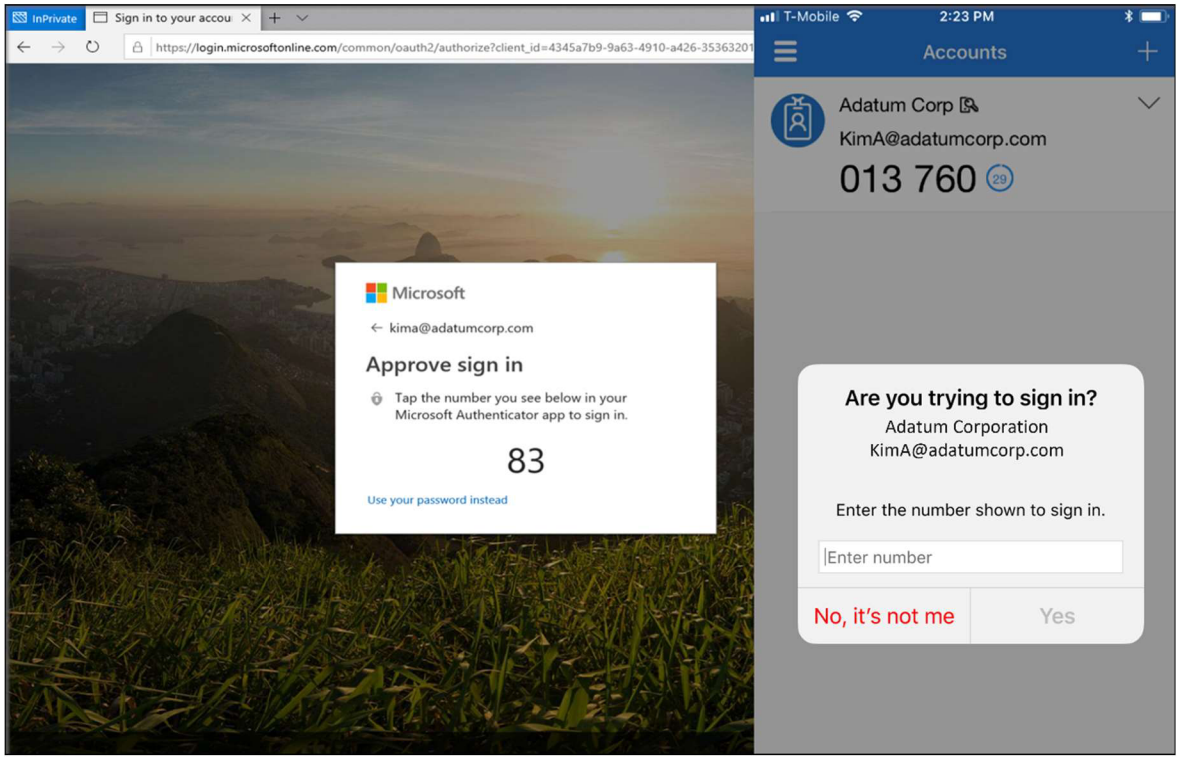


Obrázek 1 - Přihašování na základě uživatelkého jména a hesla, zdroj: vlastní tvorba z webu is.czu.cz

3.1.2 Multifaktorová autentizace

Multifaktorová autentizace (MFA) využívá více faktorů k ověření uživatele. Nejčastější je autentizace dvoufaktorová (2FA) využívaná jako doplněk k autentizaci založené na hesle. Po splnění první vrstvy zabezpečení (zadání správného uživatelského jména a hesla) je po uživateli vyžádán druhý ověřovací faktor. Obvykle se jedná o kód odeslaný na e-mail, případně v SMS zprávě.

Stále častější jsou autentizační aplikace, které generují časově omezené jednorázové kódy. V případě, že je aplikace nainstalována na mobilním telefonu, je přístup do ní mnohdy chráněn biometricky, pokud dané zařízení touto technologií disponuje.(2)(3)(6)



Obrázek 2 - Mutifaktorová autentizace(7)

3.1.3 Biometrická autentizace

Biometrická autentizace využívá jedinečné znaky uživatele. Přístupovým klíčem se tak stává samotný uživatel, přesněji jeho fyziologické či behaviorální vlastnosti. Na začátku je vždy nutné vytvořit vzor se kterým bude porovnávána předložená charakteristika.

V případě fyziologických vlastností je cílem najít jedinečné a opakovatelně identifikovatelné vlastnosti osoby, jako například otisk prstu, rozpoznání obličeje, či rozpoznání sítnice. Ověření na základě behaviorálních vlastností je pak založeno na jedinečných a opakovatelných zvycích dané osoby, jako například hlas, nebo podpis. U

behaviorální charakteristiky je vetší jistota, že autentizaci provádí opravdu daná osoba. Příkladem může být využití hlasu. Pokud bude osoba donucena k autentizaci, může cíleně hlas měnit tak, aby nebyl rozpoznán. V případě autentizace typicky fyziologické, například otiskem prstu, nemůže při donucení autentizaci nijak zabránit. Přiložený prst se bude shodovat se vzorem v databázi.

Nespornou výhodou biometrické metody autentizace je fakt, že autentizační údaje není možné zapomenout a ani si je uživatel nikam nepoznamenává. (8)(9)(10)

Obrázek 3 - Biometrická autentizace(11)

3.1.4 Single Sign-On

Metoda jednotného přihlášení umožňuje uživateli využít jednu sadu přihlašovacích údajů pro ověření uživatele na větším počtu míst. Údaje uživatele jsou uloženy u poskytovatele SSO, který ověření zajišťuje a spolupracujícím entitám potvrzuje, zda se u něj uživatel úspěšně ověřil. Pokud ano, je mu udělen přístup i do prostorů spolupracující entity.

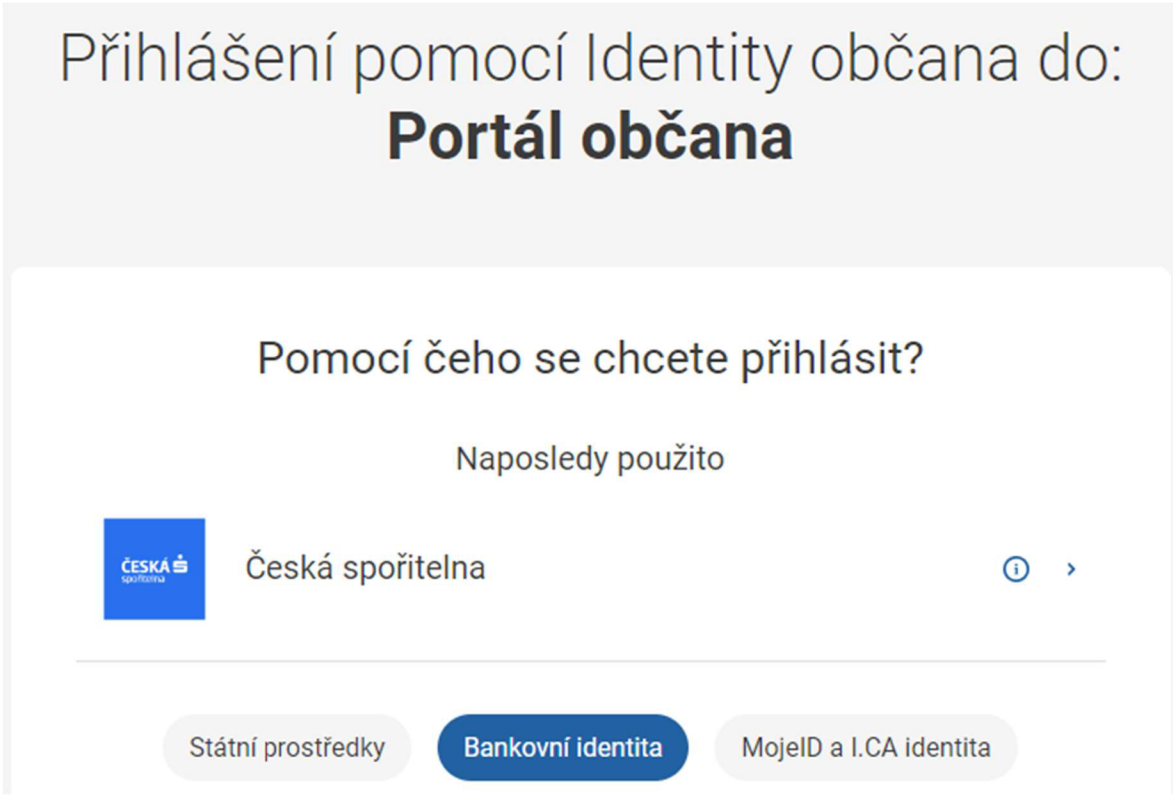
Nespornou výhodou pro uživatele je, že mu stačí pamatovat si jednu sadu uživatelských údajů. Čím nižší je počet míst, kde se citlivé údaje nachází, tím jsou bezpečněji uložené. Poskytovatel SSO je obvykle důvěryhodná společnost s velkými kapitálem. Může tak investovat nemalé částky k ochraně přístupových údajů před kybernetickými útoky.

Mezi nejznámější SSO řešení je účet Microsoft, přihlášení přes Google, Apple ID.(12)(6)(3)

Obrázek 4 - Přihlášení pomocí SSO(13)

3.1.5 Bankovní identita

Bankovní identita je metoda digitálního ověření totožnosti poskytované bankami. V České republice je pod značkou Bank iD k dispozici od ledna 2021. Banky jsou, vzhledem k jejich zabezpečení a jejich regulacím, důvěryhodným uchovatelem osobních údajů. Stát zabezpečení Bank iD důvěřuje natolik, že ho lze využít pro přihlášení k internetovým službám vystaveným státními aparáty. (14)



Obrázek 5 - Přihlášení pomocí bankovní identity(15)

3.1.6 Autentizace pomocí tokenu

Autentizace pomocí tokenu závisí na autentizačním tokenu, kterým je obvykle fyzická věc (přihlašovací karta, mobilní telefon), která obsahuje přihlašovací údaje uživatele. Autentizace je bezpečná, dokud fyzický token nepadne do nepovolaných rukou.

Tento typ autentizace je často používán v regulovaných odvětvích – bankovnictví, pojišťovnictví, zdravotnictví. Propojení tokenu s dalšími metodami zvyšuje míru zabezpečení a snižuje riziko zneužití tokenu.

Dalšími metodami, jak zajistit, že se token nedostal do nesprávných rukou, je kontrola, zda je používán stále stejně. Je obvyklé, že pokud je token odcizen, bude použit na neobvyklých místech v neobvyklý čas.(16)

Obrázek 6 - Generátor tokenů(17)

3.1.7 Certifikátová autentizace

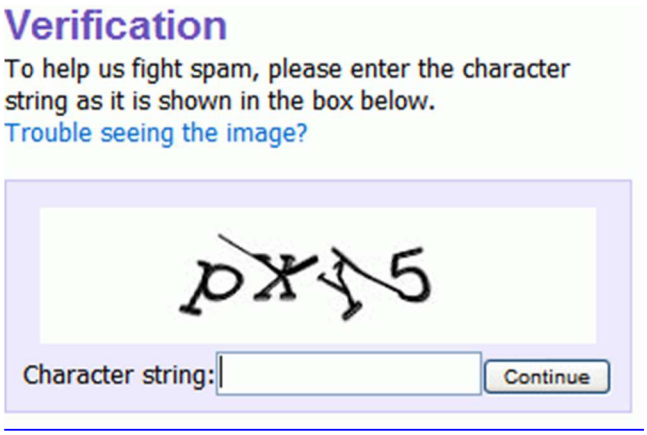
Uživatele lze ověřit i pomocí digitálních certifikátů. Certifikáty obsahují veřejný klíč spojený s identitou uživatele. Certifikáty jsou vydávány důvěryhodnými certifikačními autoritami. Pro komunikaci s úřady v České republice jsou to První certifikační autorita, PostSignum a eDentity. (18)

3.2 CAPTCHA

CAPTCHA (completely automated public Turing test to tell computers and humans apart) je technologie, jejíž cílem je odlišit skutečného lidského uživatel od automatizovaného programu – bota. Důvodem, proč nejsou boti na stránkách vítáni, je jejich pochybná aktivita, především zneužívání webových formulářů a opakované pokusy o přihlášení, kterými se pokouší zjistit přístupové údaje uživatelů. Pro rozlišení bota a skutečného člověka využívá CAPTCHA úlohy, které jsou jednoduše splnitelné pro člověka, ale složité pro bota.

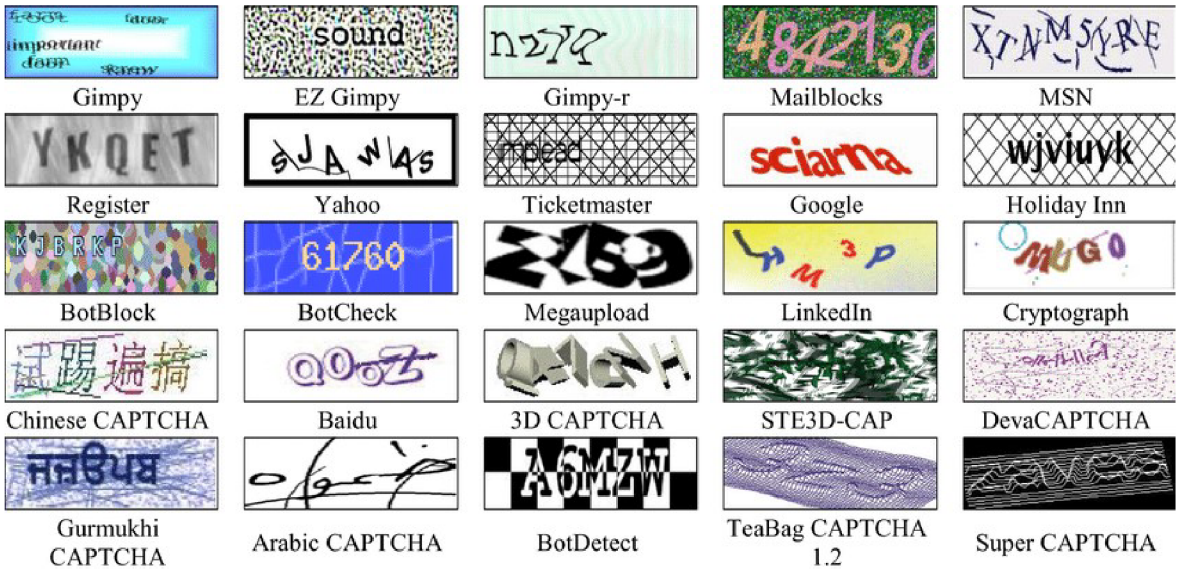
3.2.1 Historie vzniku CAPTCHA

Vznik CAPTCHA se datuje na rok 2000, kdy byla vyvinuta týmem programátorů na Carnegie Mellon University. Hlavní myšlenkou bylo zabránit botům, kteří zahlcují webové stránky nesmyslnými registracemi. První stránkou, která CAPTCHA implementovala bylo v roce 2001 YAHOO!. Při počátku CAPTCHA byl test založen na rozluštění pokřiveného textu, což časem přestalo stačit. Boti se naučili text rozpoznávat a zároveň nebyl test založený pouze na zraku vhodný pro zrakově postižené uživatele. (19; 1)



Obrázek 7 - Textová CAPTCHA(20)

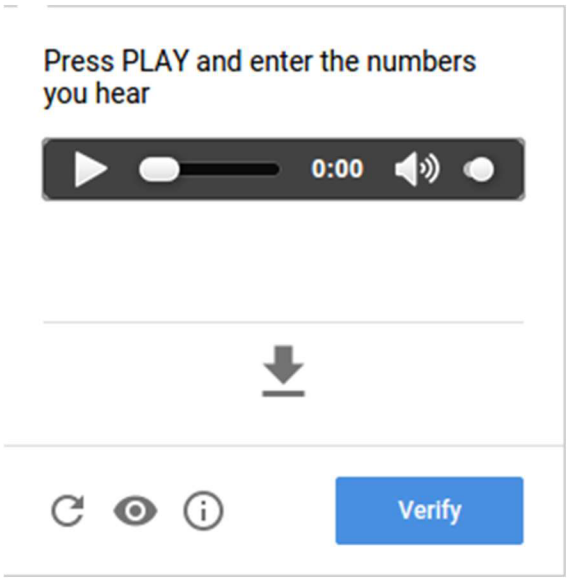
První CAPTCHA představovala jednoduchou úlohu, která spočívala v rozpoznání deformovaného textu a jeho přepis do pole. Jedná se o nejstarší metodu, na kterou už jsou boti připravení a zvládnou ji vyřešit.(19)



Obrázek 8 - Příklady textových CAPTCHA(21)

3.2.2 Audio CAPTCHA

Vzhledem k nevhodnosti textové CAPTCHA pro zrakově postižené vznikla verze založená na zvuku. Uživatel je vyzván k poslechu nahrávky, která obsahuje kód či jiné instrukce, jak testem projít. Vzhledem k tomu, že i tento typ je boty překonatelný, jsou do nahrávky přidávány různé hluky a šumy, které mají botům rozluštění ztížit.(4)(1)



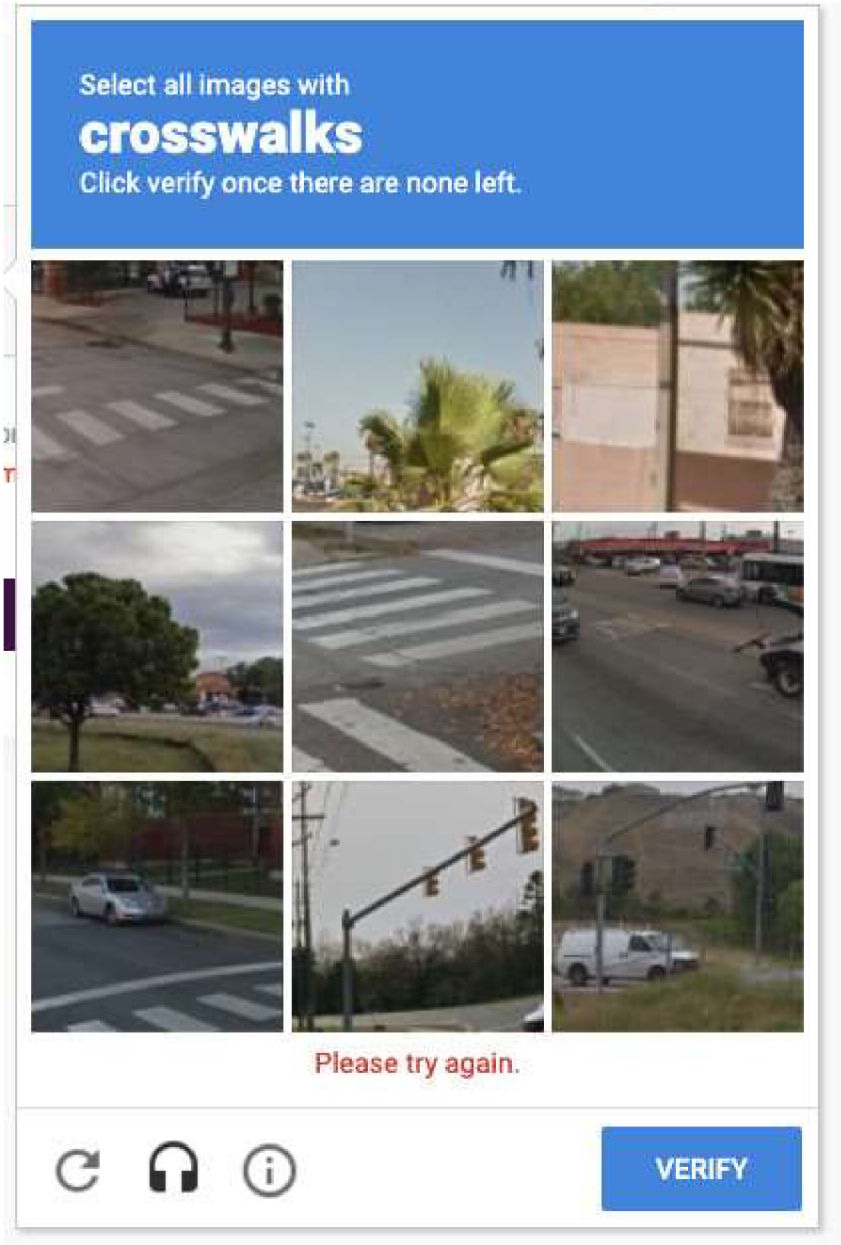
Obrázek 9 - Audio CAPTCHA(22)

3.2.3 Obrázková CAPTCHA

Obrázková CAPTCHA je založena na identifikaci obsahu obrázku

Nejčastěji se může uživatel setkat s rozpoznáváním vzorů, nebo s výběrem. Typickým úkolem je vybrat z předložených obrázků takové, které obsahují určenou věc, například dopravní značku.

Vzhledem k tomu, že tyto základní metody založené na výběru jsou řešitelné boty, přichází nové typy úkolů, jako například otočit obrázek tak, aby byl správně orientován, nebo posunout dílek obrázku tak, aby do obrázku zapadl.



Obrázek 10 - Obrázková CAPTCHA(23)

3.3 ReCAPTCHA

Speciálním typem CAPTCHA je její pokročilá verze ReCAPTCHA vlastněná společností Google. Myšlenka ReCAPTCHA se shoduje s myšlenkou této diplomové práce – využít snahu uživatelů při plnění testu k něčemu jinému. Poprvé byla využita pro digitalizaci archivu New York Times. Slova, která nedokázal program pro rozpoznání optických znaků, byla předložena jako zadání pro uživatele ReCAPTCHA. Aby bylo zajištěno, že si uživatel nevymýšlí a nevkládá náhodné znaky, bylo nerozpoznané slovo předkládáno společně s již ověřeným rozpoznaným slovem a zároveň bylo předloženo většímu počtu uživatelů.

Pomocí ReCAPTCHA byl archiv New York Times, archiv novin vydávaných od roku 1851, již plně digitalizován. Nejtěžší bylo rozluštit výtisky vydané před rokem 1900, neboť jejich obraz obsahoval různé šmouhy, fleky a jiné nedokonalosti. Problém také tvořila slova, která v době digitalizace nebyla v anglickém slovníku.

Pro podobné účely, a sice digitalizace textu, který je těžce čitelný a který OCR (Optical Character Recognition) nedokázal obstojně rozluštit, byla ReCAPTCHA využita i ve službě Google Books.(19; 24; 25)(24)(25)

Obrázek 11 - Původní textová ReCAPTCHA(25)

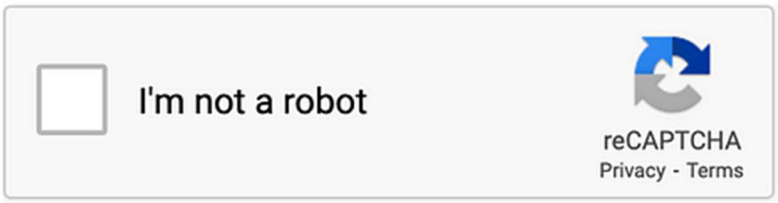
3.3.1 Vývoj ReCAPTCHA

První verze ReCAPTCHA, která pomohla digitalizovat archiv New York Times a byla využívána jako lidmi asistované OCR, byla v roce 2012 upravena a uživatelům byly začaly být předkládány obrázky z Google Street View. Jedná se o službu společnosti Google, která pořizuje snímky ulic reálného světa. ReCAPTCHA zde byla využita pro identifikaci obsahu pořízených obrázků. Úkolem byla identifikace například chodců, přechodů, semaforů či automobilů. Takto anotované obrázky by byly skvělým zdrojem dat pro autonomní automobily. Existovala hypotéza, že Google obrázky tímto způsobem využívá ve svém

projektu, který měl za cíl tvorbu autonomního vozu. Google vyvrátil toto použití a tvrdí, že data byla použita pouze pro vylepšení služby Google Maps.

V roce 2014 vydal Google “no CAPTCHA ReCAPTCHA” která po uživateli nepožaduje splnění testu, ale pouze zaškrtnutí zaškrtávacího políčko. No CAPTCHA se rozhoduje na základě chování uživatele na webových stránkách,

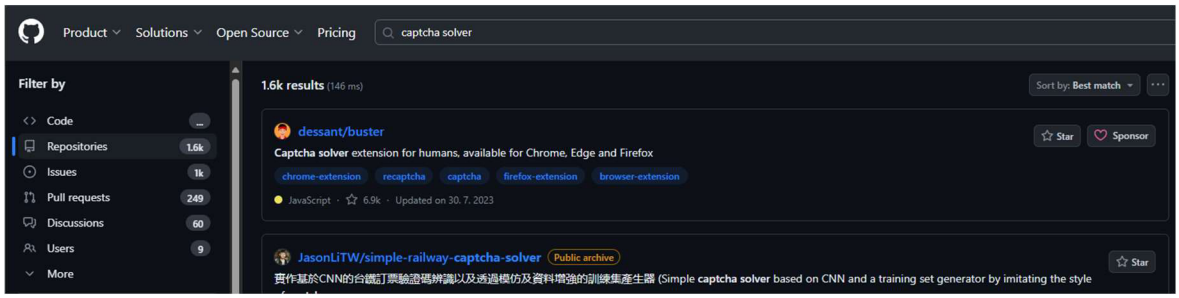
Netrvalo to dlouho a v roce 2017 představil Google neviditelnou ReCAPTCHA, která kompletně probíhá na pozadí a pokud není uživatel podezřelý, nemusí plnit žádný test, ani zaškrtávat políčko.(19)(25)(24; 26)



Obrázek 12 - NoCAPTCHA(25)

3.3.2 Automatické řešení ReCAPTCHA

Aby bylo možné i přes existující ReCAPTCHA boty používat, existuje mnoho druhů programů, které CAPTCHA automaticky řeší. K dispozici jsou placené, ale i volně dostupné verze. Programy nemusí používat pouze boti. Některé uživatele vyplňování CAPTCHA obtěžuje a zdržuje. Jen na GitHubu se po vyhledání slovního spojení captcha solver objeví více než 1,5 tisíc repositářů.(27)(28)



Obrázek 13 - Repositář GitHub(29)

3.4 Anotace dat

Anotace dat je proces, při kterém jsou data označována, popisována, jsou jim přidávána metadata. Cílem anotace je lépe porozumět obsahu dat, popsat je, jsou anotována, aby je počítače dokázaly lépe interpretovat a pochopily je. Může se jednat o téměř jakýkoliv typ dat – text, video, audio, nebo obrázky. Anotace může probíhat manuálně, data popisuje člověk, nebo automaticky využitím algoritmů strojového učení. Anotované datasety jsou nezbytné pro tvrobu modelů strojového učení, protože model musí pochopit vstupní vzory, aby je mohl zpracovat a předpovědět přesný výsledek. Kvalita modelu závisí na kvalitě a kvantitě dat použitých při jeho tvorbě. Prvním krokem při tvorbě modelů pro umělou inteligenci je tedy hledání dostatečného množství anotovaných dat.

Anotované datasety je možné získat zdarma z mnohých internetových stránek, avšak pokud se jedná o specifická interní data, nad kterými chceme vytvořit model, je nutné si anotaci zařídit interně, nebo si jí objednat u externí firmy.(30)

3.4.1 Reinforcement learning from human feedback

Metoda RLHF (Reinforcement learning from human feedback) využívá k poučení se zpětnou vazbou od lidí. Stala se velmi populární po úspěchu velkých jazykových modelů. Lidé v tomto procesu vytvářejí vhodné odpovědi, nebo vybírají lepší z odpovědí, které jim byly nabídnuty. Lidská práce je však nákladná, a proto existuje možnost RLAIF (Reinforcement learning from AI feedback) kde je k poučení využita dobře natrénovaná AI s vysokou úspěšností.(31)

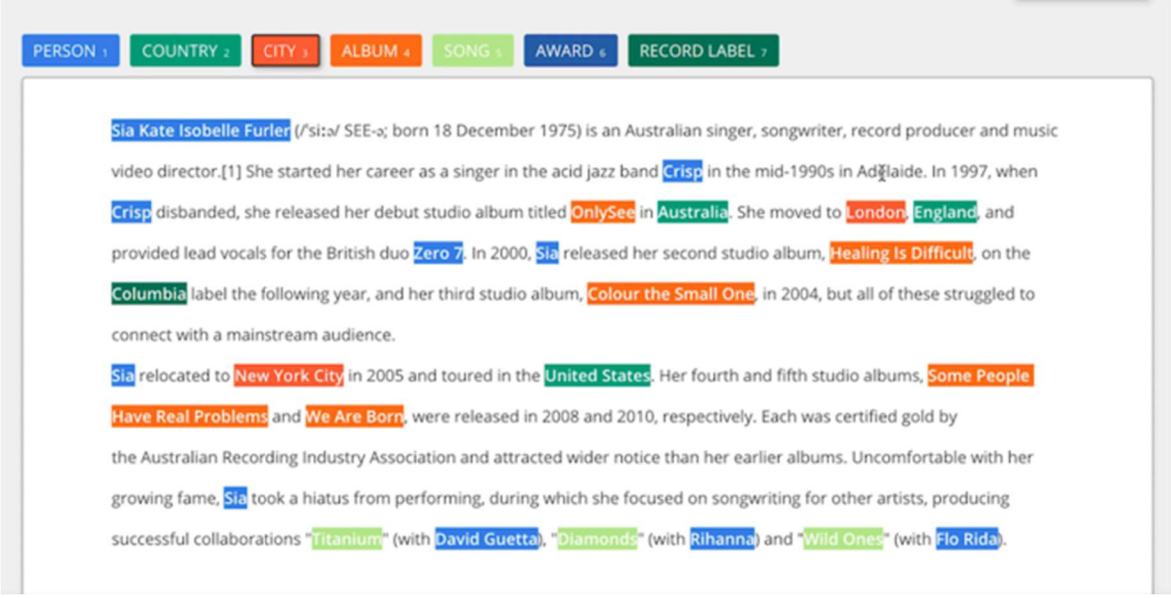
3.4.2 Anotace textu

Anotace textu je využita pro lepší porozumění textu stroji. Pokud je text správně anotován, pomáhá například chatbotům najít správnou odpověď na požadavek uživatele. Text lze anotovat více způsoby.

Na základě sémantiky je text popsán tak, aby bylo možné lépe porozumět jeho obsahu a významu. Lze přiřazovat různé kategorie, klíčová slova, či vztahy mezi entitami.

Dále lze v textu pozorovat a anotovat konkrétní záměry nebo cíle uživatele, například požadavek “Spojit s operátorem”, proto, aby byl model schopný záměry rozpoznat a reagovat na ně.(32)(33)

Zajímavá je také anotace na základě sentimentu, emocí. Části textu mohou být označeny jako pozitivní, neutrální, nebo negativní v závislosti na emocionálním obsahu, díky čemuž pak může strojový model porozumět emocionálnímu kontextu textu.

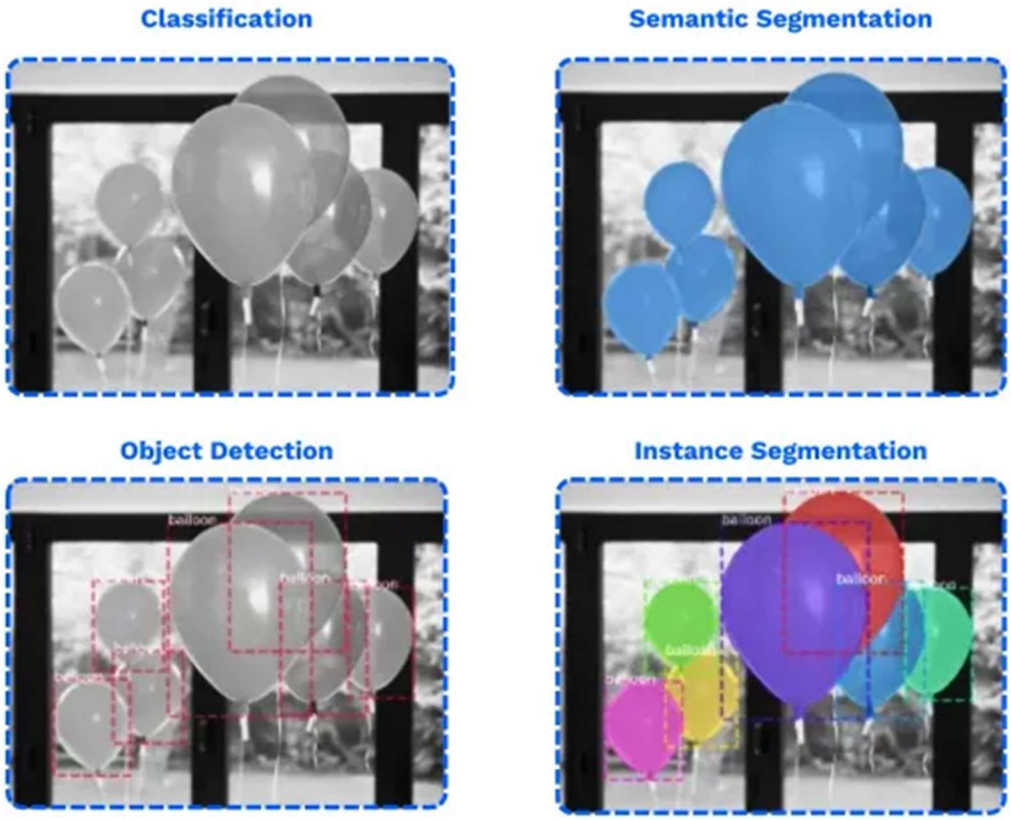


Obrázek 14 - Anotace textu(33)

3.4.3 Anotace obrázku

Základní anotování je na základě obsahu obrázku. Jedná se o jeden popis obsahu obrázku, například “auto”. Model naučený na takto anotovaných datech zvládá klasifikovat obrazová data a rozhodnout, co se na obrázku nachází.

Složitější a přesnější je anotace objektů na obrázku. Díky přesnější anotaci by měl být model schopný vypsat jednotlivé entity, které je naučený v obrázku hledat.(33)(34)(30)



Obrázek 15 - Anotace obrázku(35)

3.4.4 Ostatní anotace

Anotovat lze i video a zvuk. Vždy je cílem připravit data pro učení strojového modelu. Ve videu lze rozpoznávat objekty, zvuk lze popsat nebo z něj vysledovat barvu hlasu, emoce.

Existují také specifické anotace dat pro lékařství, automobilový průmysl, strojní průmysl a mnoho dalších.(30)(35)

Obrázek 16 - Anotace videa(35)

4 Vlastní práce

Na základě analýzy stávajících řešení v oblasti anotace dat a ověření uživatele je vlastní práce věnována návrhu a naprogramování aplikace podobná ReCAPTCHA. Aplikace bude využívat odpovědi uživatelů k anotaci dat. Výhodou aplikace je, že data anotuje pro nás, nikoliv pro třetí subjekt, který by nám CAPTCHA řešení dodával.

4.1 Návrh aplikace

Aplikace se svou funkcionalitou inspiruje u již vyřazené 1. verze ReCAPTCHA. Vstupem do aplikace jsou obrázky, které je třeba anotovat. Obrázky jsou uloženy do složek, které nesou název hledaného objektu. Každá taková složka obsahuje 2 podsložky "true", obsahující obrázky s ověřeným výskytem hledaného objektu, a "unknown" obsahující obrázky, pro které je výskyt ověřován. Test se skládá z 9 obrázků, vždy 3 ve 3 řadách. Otázka je vždy položena kladně, například: "Označte obrázky, na kterých se vyskytuje vozidlo". V každém testu se vždy nachází minimálně 1 obrázek z množiny true, který musí uživatel označit, aby byl jeho pokus označen jako úspěšný. Jedná se o testovací množinu, která zajišťuje, aby test nepropustil každého bez ohledu na označená pole a snahu označit vyhledávaný objekt. Pokud je uživateli předložen větší počet obrázků z množiny true, musí pro úspěšné ověření označit všechny. V případě neúspěšného pokusu je o této skutečnosti uživatel informován a je mu test s novými daty předložen znovu. Pokusy uživatelů, ať úspěšné či neúspěšné, jsou zaznamenány pro vyhodnocení a anotaci dat ve složce unknown.

Aplikace je naprogramována v programovacím jazyku Python a využívá volně stahovatelné balíčky pro tento jazyk. Pro tvorbu testovací stránky je využito HTML společně s CSS a Javascript.

Důležité proměnné jsou uloženy v konfiguračním souboru a je možné je jednoduše upravit je dle potřeby.

4.1.1 Příprava dat

Aplikace mimo modul CAPTCHA obsahuje i modul pro přípravu dat, který kontroluje a zajišťuje správnou strukturu složek a předkládaných obrázků. Složky nesou název hledaného objektu a obsahují podsložky true a unknown. Obrázky mají stejnou velikost.

4.1.2 Vyhodnocení dat

Aplikace obsahuje také modul pro vyhodnocení dat, který pracuje se záznamy z CAPTCHA. Modul na základě získaných dat anotuje obrázky z předložených složek. A ukládá je do roztříděné jako true, false a unknown.

4.2 Použitý software

4.2.1 Python

Pro naprogramování aplikace je vybrán jazyk Python jedná se o moderní objektově orientovaný programovací jazyk. Je známý svou čitelností a jednoduchostí syntaxe, díky čemuž je ideálním jazykem pro začátečníky. Python má rozsáhlou komunitu, která tvoří volně stahovatelné knihovny, které obsahují užitečné funkce.

4.2.2 Flask

Flask je jednou z knihoven, kterou je možné do Pythonu stáhnout. Jedná se o framework pro tvorbu webových aplikaci. Nabízí základní funkce pro správu šablon a integraci s databázemi společně s API. Je snadné ho rozšiřovat pomocí dalších knihoven.

4.2.3 HTML

HTML (HyperText Markup Language) je základním stavebním kamenem webových stránek. Vzhledem k tomu, že CAPTCHA bude využívána jako ověření pro uživatele, je třeba připravit webovou stránku pro otestování funkčnosti aplikace. HTML tvoří statické stránky, které se interakcí uživatele nemění, a proto je třeba využít pro pokročilé funkce například Javascript.

4.2.4 Javascript

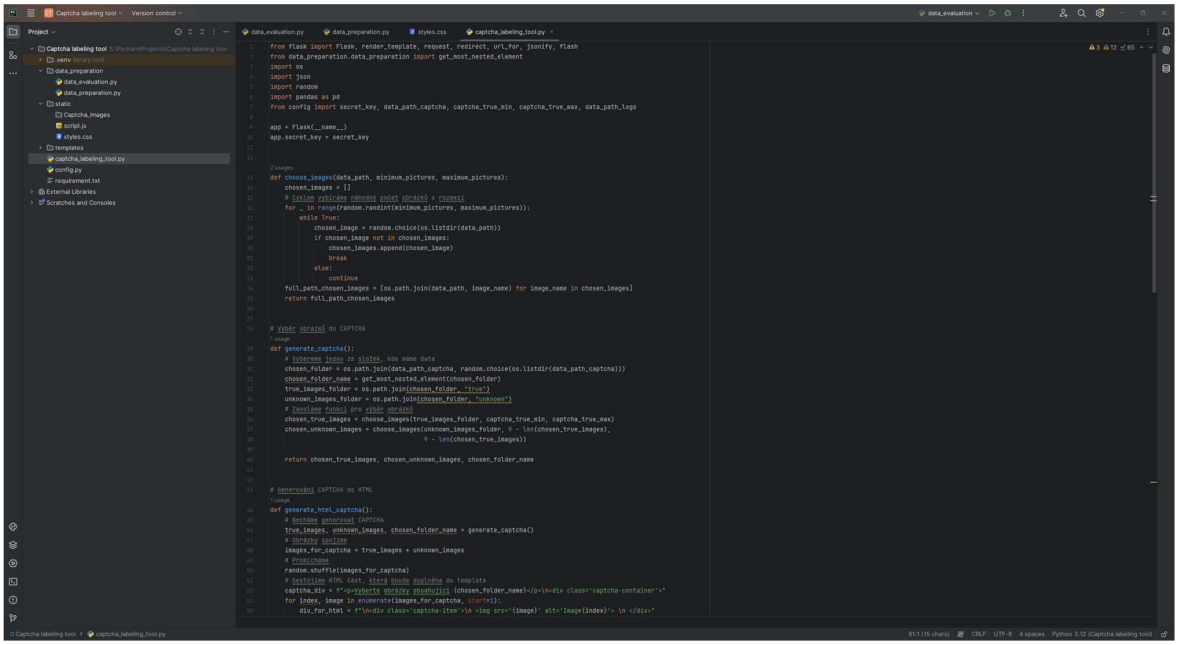
Pro tvorbu některých funkcionalit na webové stránce na straně uživatele je vybrán skriptovací jazyk Javascript. Umožňuje manipulaci s obsahem HTML a CSS společně s detekcí událostí a reakcí na ni.

4.2.5 CSS

CSS (Cascading Style Sheets) je nezbytné pro úpravu vzhledu a formátování HTML. Umožňuje definovat styly a vizuální vlastnosti HTML.

4.2.6 PyCharm

Aplikace byla vyvíjena ve vývojovém prostředí PyCharm, které je vyvíjeno českou společností JetBrains s.r.o. Prostředí je uživatelsky přívětivé, přehledné, moderní a velký počet funkcí usnadňuje vývoj softwaru.



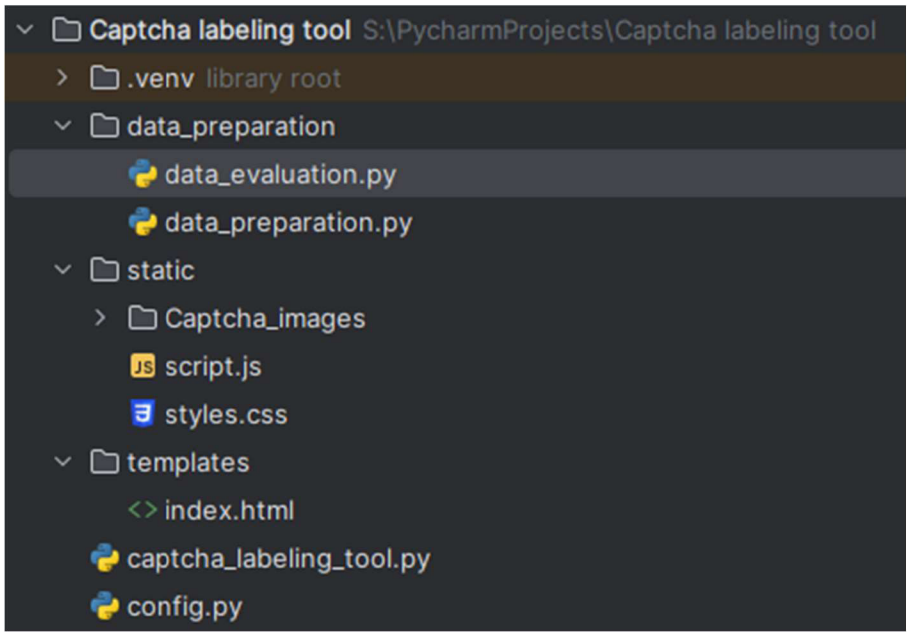
Obrázek 17 - Vývojové prostředí PyCharm, zdroj: vlastní tvorba

4.3 Vývoj aplikace

Aplikace je vyvíjena ve vývojovém prostředí PyCharm v Pythonu verze 3.12. Je vybudováno virtuální prostředí, ve kterém budou pouze balíčky, které se stáhnou při programování aplikace.

4.3.1 Modul pro přípravu dat

Na základě návrhu aplikace je nejdříve naprogramován modul pro přípravu dat. Nachází se ve složce data_preparation v souboru data_preparation.py a jeho cílem je připravit data do podoby, aby mohly být použity v testu CAPTCHA.



Obrázek 18 - Repozitáč aplikace, zdroj: vlastní tvorba

Složky s připraveným datasetem pro anotaci jsou uloženy do složky 01_raw v adresáři zvoleném v konfiguračním souboru. Název složky je roven hledanému objektu. Pro příklad, máme-li dataset, ve kterém chceme zjistit, zda se na obrázcích vyskytuje motocykl, je název složky "motocykl". Z názvu složky je skládána otázka do testu CAPTCHA.

Každá taková složka obsahuje podsložku "true" a "unknown". Složka true obsahuje obrázky s ověřeným výskytem hledaného objektu. V našem příkladě by se jednalo o obrázky motocyklů. Složka unknown obsahuje obrázky, které je třeba anotovat a u kterých výskyt objektů potvrzený nemáme.

Data jsou připravována funkcí preprocess_raw_folders. Jsou zjištěny názvy složek ve složce 01_raw a dále zpracovány for cyklem. Pro každou složku je volána funkce

check_structure, která kontroluje, zda složka obsahuje vše, co má. Kontrolována je existence podložek true a unknown a jejich obsah. Aby byla složka vyhodnocena jako správně strukturovaná, musí obsahovat minimálně 10 příkladů ověřených výskytů ve složce true a 20 obrázků k ověření ve složce unknown. Tyto limity jsou změnitelné v konfiguračním souboru. Pokud složka projde základní kontrolou, je vyrobena její kopie ve složce 02_processed i s podložkami, avšak bez obrázků.

```
#Příprava obrázků ze složky 01_raw
1 usage
def preprocess_raw_folders():
    for folder_name in [item for item in os.listdir(data_path_raw) if os.path.isdir(os.path.join(data_path_raw, item))]:
        raw_folder_path = os.path.join(data_path_raw, folder_name)
        raw_folder_path_true = os.path.join(raw_folder_path, "true")
        raw_folder_path_unknown = os.path.join(raw_folder_path, "unknown")
        # Voláme kontrolu struktruy složek
        check_structure(raw_folder_path)

        #Kontrola, zda už složku v připravených nemáme
        processed_folder_path = os.path.join(data_path_processed, folder_name)
        if os.path.exists(processed_folder_path):
            raise Exception(f"Složka {folder_name} již existuje v adresáři připravených složek.")

        os.makedirs(processed_folder_path)

        processed_folder_path_true = os.path.join(processed_folder_path, "true")
        processed_folder_path_unknown = os.path.join(processed_folder_path, "unknown")

        os.makedirs(processed_folder_path_true)
        os.makedirs(processed_folder_path_unknown)

        #Připravuejem obrázky v test složce
        for index, file in enumerate(os.listdir(raw_folder_path_true)):
            file_path = os.path.join(raw_folder_path_true, file)
            prepare_picture(processed_folder_path_true, file_path, "t" + str(index))
        # Připravuejem obrázky v unknown složce
        for index, file in enumerate(os.listdir(raw_folder_path_unknown)):
            file_path = os.path.join(raw_folder_path_unknown, file)
            prepare_picture(processed_folder_path_unknown, file_path, "u" + str(index))
```

Obrázek 19 - Funkce preprocess_raw_folders, zdroj: vlastní tvorba

Obrázky jsou zpracovány funkcí prepare_picture. Jako ideální velikost obrázků je zvolena 100px na 100px. Obrázkům je změněna velikost. Ideální jsou obrázky, které jsou původně 1 ku 1. Pokud tomu tak není, nevadí, aplikace i tak vytváří obrázky o velikosti 100px na 100px, obsahují však bílé okraje. Obrázky jsou přejmenovány. Název obrázků ve složce true je tvořen písmenem “t” a číslem pořadí obrázku, v jakém byl funkcí prepare_picture zpracován. Obdobně je tomu pro obrázky ve složce uknown, akorát že zde je použito písmeno u. Tímto pojmenováním je zajištěno, že při programování a odstraňování chyb vývojář z názvu pozná, o který typ obrázku se jedná. Takto upravené obrázky jsou ukládány do připravených prázdných složek v adresáři 02_processed. Tím je proces přípravy dat pro danou složku ukončen a nastupuje jiná složky s jiným názvem a daty k anotaci.

```python
#Připraví obrázky pro CAPTCHA
2 usages
def prepare_picture(output_folder, file, file_name):
    try:
        #Zvolíme veliost
        size = image_size
        with Image.open(file) as img:
            img.thumbnail(size, Image.LANCZOS)
            # Bílý podklad
            new_img = Image.new( mode: "RGB", size, color: "white")
            left = (size[0] - img.width) // 2
            top = (size[1] - img.height) // 2
            right = left + img.width
            bottom = top + img.height
            # Vloží obrázek doprostřed bíleho podkladu
            new_img.paste(img, box: (left, top, right, bottom))
            new_img.save(os.path.join(output_folder, file_name + ".jpg"))
    except:
        raise Exception(
            f"Chyba v přípravě obrázku {get_most_nested_element(file)} ze složky folder {get_most_nested_element(output_folder)}")
```

Obrázek 20 - Funkce prepare_picture, zdroj: vlastní tvorba

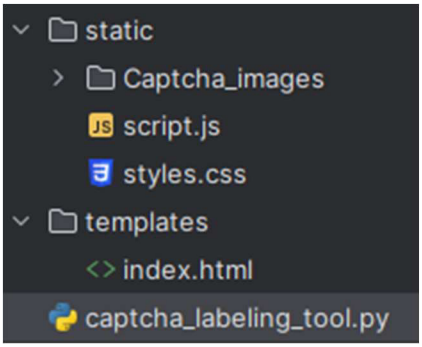
Po přípravě všech složek z adresáře 01_raw funkcí preprocess_raw_folders je volána funkce move_processed_folders_to_captcha, která zajišťuje přesun připravených složek z 02_processed do adresáře static uvnitř aplikace CAPTCHA proto, aby jej aplikace mohla začít předkládat uživatelům v testech.

```python
# Přesun připravené složky do úložiště CAPTCHA
1 usage
def move_processed_folders_to_captcha():
    for folder in os.listdir(data_path_processed):
        captcha_path = os.path.join(os.path.dirname(os.getcwd()), data_path_captcha)
        shutil.move(os.path.join(data_path_processed, folder), captcha_path)
```

Obrázek 21 - Funkce move_processed_folders_to_captcha, zdroj: vlastní tvorba

4.3.2 Modul CAPTCHA

Po přípravě dat je čas na programování samotného modulu CAPTCHA. Aby bylo možné uživatelům test předkládat a otestovat jej, je pro tyto potřeby naprogramována jednoduchá webová stránka. Obsahuje pouze jednu stránku index.html, na kterou je CAPTCHA generována. Stránka obsahuje tlačítko pro potvrzení odeslání CAPTCHA a instrukci k vyplnění. Aby se mřížka, která CAPTCHA obsahuje správně zobrazovala, je na stránku navázáno CSS (Cascading Style Sheets) styles.css obsahující základní nastavení. Na straně aplikace běží webový framework flask, který se stará o komunikaci mezi apliacká a uživatelem.



Obrázek 22 - Soubory pro modul CAPTCHA, zdroj: vlastní tvorba

Pokud se na volání funkcí podíváme od té nejvíce zanořené, začíná vše výběrem obrázků. Z adresáře static, z podsložky Captcha_images je náhodně vybrána jedna složka s připravenými datasety k ověření. Z této složky je následně vybrán náhodný počet obrázků kontrolních ze složky true, aktuálně je nastaveno mezi 1 až 4 obrázky, a zbylý počet do 9 obrázků ze složky unknown. Celou dobu se pracuje s celou cestou k obrázku, neboť ta bude předkládána do HTML. Po čas zpracování je předáváno jméno složky, neboť je použito do otázky a listy s obrázky množiny true a obrázky množiny unknown.

```
# Výběr obrázků do CAPTCHA
1 usage
def generate_captcha():
    # Vybereme jednu ze složek, kde máme data
    chosen_folder = os.path.join(data_path_captcha, random.choice(os.listdir(data_path_captcha)))
    chosen_folder_name = get_most_nested_element(chosen_folder)
    true_images_folder = os.path.join(chosen_folder, "true")
    unknown_images_folder = os.path.join(chosen_folder, "unknown")
    # Zavoláme funkci pro výběr obrázků
    chosen_true_images = choose_images(true_images_folder, captcha_true_min, captcha_true_max)
    chosen_unknown_images = choose_images(unknown_images_folder, 9 - len(chosen_true_images),
                                          9 - len(chosen_true_images))

    return chosen_true_images, chosen_unknown_images, chosen_folder_name
```

Obrázek 23 - Funkce generate_captcha, zdroj: vlastní tvorba

V posledním kroku uvnitř funkce generate_html_captcha jsou obrázky spojeny, zamíchány a připraveny do HTML kódu ve vlastních elementech společně s otázkou, založenou na jméně složky. Název složky se do otázky promítne vždy na konec za "Vyberte obrázky obsahující". Ku příkladu, je-li název složky "motocykl", je položena otázka: "Vyberte obrázky obsahující motocykl". Funkce vrací jak připravenou část HTML kódu, tak jméno složky a oba listy (true_images a unknown_images).

```python
# Generování CAPTCHA do HTML
1 usage
def generate_html_captcha():
    # Necháme generovat CAPTCHA
    true_images, unknown_images, chosen_folder_name = generate_captcha()
    # Obrázky spojíme
    images_for_captcha = true_images + unknown_images
    # Promícháme
    random.shuffle(images_for_captcha)
    # Sestojíme HTML část, která bnude doplněna do template
    captcha_div = f"<p>Vyberte obrázky obsahující {chosen_folder_name}</p>\n<div class='captcha-container'>"
    for index, image in enumerate(images_for_captcha, start=1):
        div_for_html = f"\n<div class='captcha-item'>\n <img src='{image}' alt='Image{index}'> \n </div>"
        captcha_div += div_for_html

    captcha_div += "\n </div>"
    return captcha_div, true_images, unknown_images, chosen_folder_name
```

Obrázek 24 - Funkce generate_html_captcha, zdroj: vlastní tvorba

Připravená část HTML kódu je přidána do index.html, čímž je vygenerována CAPTCHA a stránka je metodou GET předána uživateli.

```python
# Flask
@app.route( rule: '/', methods=['GET', 'POST'])
def index():
    global true_images, unknown_images, chosen_folder_name
    # Generování CAPTCHA a předložení uživateli
    if request.method == 'GET':
        captcha, true_images, unknown_images, chosen_folder_name = generate_html_captcha()
        return render_template( template_name_or_list: 'index.html', captcha=captcha)
```

Obrázek 25 - Metoda GET, zdroj: vlastní tvorba

Uživatel dle zadání vybírá obrázky, na kterých se hledaný objekt nachází. Navázaný Javascript (script.js) zajišťuje pomocí funkce toggleImageSelection označení obrázků, které jsou užviatelem zvoleny. Takové obrázky mají modré okraje. Po stisknutí tlačítka pro odeslání jsou metodou POST získány vybrané obrázky.

```python
# Příjem odpovědi od uživatele
elif request.method == 'POST':
    selected_images = request.form.get('selected_images')
    try:
        selected_images = json.loads(selected_images)
        # Kontrola správné odpovědi a logování odpovědi
        correct_selection = all(image_path in selected_images for image_path in true_images)
        if correct_selection:
            log_result(chosen_folder_name, status: 'success', true_images, unknown_images, selected_images)
            return jsonify({'status': 'success', 'message': 'Vybrány správné obrázky!'})
        else:
            log_result(chosen_folder_name, status: 'error', true_images, unknown_images, selected_images)
            # Posíláme uživateli hlášku, že odpověděl nesprávně
            flash( message: 'Vybrány nesprávné obrázky! Zkuste to znovu.', category: 'error')
            return redirect(url_for(".index"))
```

Obrázek 26 - Metoda POST, zdroj: vlastní tvorba

Obrázky, které vybral uživatel, jsou porovnány s obrázky true a pokud jsou označeny všechny, vrací se uživateli zpráva, že byl úspěšný. Pokud úspěšný nebyl, je informován hláškou "Vybrány nesprávné obrázky! Zkuste to znovu." a je mu předložena nová CAPTCHA.

Odpověď uživatele, ať úspěšná či neúspěšná, je uložena do souboru results.json. Ukládá se název složky, ze které řešené obrázky pocházely, status zprávy (success/error), časové razítko, předložené obrázky ze složky true, předložené obrázky ze složky unknown a odpověď uživatele.

4.3.3 Modul pro vyhodnocení odpovědí

Modul pro vyhodnocení odpovědí uživatele, který provádí roztřídění předložených obrázků je stejně jako modul pro přípravu dat uložen ve složce data_preparation, avšak ve vlastním souboru data_evaluation.py.

Funkce filter_success_results vybírá z results.json odpovědi od uživatelů, které byly vyhodnocené jako úspěšné, tedy status je success.

Tyto odpovědi vstupují do funkce process_captcha_results, kde jsou na základě folder_name seskupeny odpovědi do slovníku podle tří klíčů. Unknow_tested obsahuje název testovaného obrázku a jako hodnota je uložen počet výskytu tohoto souboru v odpovědích. Unknown_true a unknown _false obsahují stejnou informaci, avšak zde je počítáno, kolikrát byl obrázek ohodnocen jako true a kolikrát jako false. Výsledkem funkce je tedy slovník, který obsahuje informace agregované dle jednotlivých názvů složek a dle ohodnocení uživatelem.

```python
#Zpracujeme výsledky
1 usage
def process_captcha_results(success_results):
    result_dict = {}
    # Projdeme každý záznam v JSON
    for item in success_results:
        folder_name = item.get('folder_name')

        if folder_name:
            unknown_images = item.get('unknown_images', [])
            selected_images = item.get('selected_images', [])

            # Inicializujeme slovníky pro 'unknown_true' a 'unknown_false' pro tento 'folder_name'
            if folder_name not in result_dict:
                result_dict[folder_name] = {'unknown_tested': {}, 'unknown_true': {}, 'unknown_false': {}}


            for image in unknown_images:
                # Připočteme si výskyt obrázku
                if image in result_dict[folder_name]['unknown_tested']:
                    result_dict[folder_name]['unknown_tested'][image] += 1
                else:
                    result_dict[folder_name]['unknown_tested'][image] = 1
                # Pokud je obrázek v 'selected_images', přidáme ho do 'unknown_true'
                if image in selected_images:
                    if image in result_dict[folder_name]['unknown_true']:
                        result_dict[folder_name]['unknown_true'][image] += 1
                    else:
                        result_dict[folder_name]['unknown_true'][image] = 1
                # Ostatní jsou 'unknown_false'
                else:
                    if image in result_dict[folder_name]['unknown_false']:
                        result_dict[folder_name]['unknown_false'][image] += 1
                    else:
                        result_dict[folder_name]['unknown_false'][image] = 1
    return result_dict
```

Obrázek 27 - Fukce preprocess_captcha_results, zdroj: vlastní tvorba

Tato data dále vstupují do funkce process_result_dict, kde je pro každý dataset vyhledán obrázek s nejmenším počtem předložení ke kontrole, což vyjadřuje číslo u obrázku v unknown_tested. Aby byl celý dataset ohodnocen, jako anotovaný a mohl pokračovat dále v procesu, musí být každý obrázek předložen ke kontrole alespoň 10krát. Tato hodnota je

upravitelná a nachází se v konfiguračním souboru jako annotation_threshold. Pokud dataset tuto podmínku nesplní, zůstává v CAPTCHA a je nadále předkládán uživatelům.

```python
# Zpracování result_dict
1 usage
def process_result_dict(result_dict):
    done_folders = []
    for folder_name, values in result_dict.items():
        # Pro každou složku vybereme počet předložení každého obrázku
        unknown_tested = values['unknown_tested']
        # Najdeme ten s nejmenším počtem předložení a porovnáme s thresholdem
        min_value = min(unknown_tested.values())
        if min_value < annotation_threshold:
            print(
                f"Složka {folder_name} obsahuje obrázky, které nesplňují minimální počet předložení ke kontrole. Složka bude ponechána v CAPTCHA")
        else:
            # Když je OK, můžeme složku poslat dále ke zpracování
            done_folders.append(folder_name)

    return (done_folders)
```

Obrázek 28 - Funkce process_result_dict, zdroj: vlastní tvorba

Název složky, která splnila podmínku je předán do funkce analyze_category_results společně s resulst_dict. Zde probíhá třídění jednotlivých obrázků dle odpovědí na true, false a unknown. Počítá se kolikrát byl obrázek označen jako true děleno kolikrát byl obrázek předložen uživatelům. Pro každý obrázek je tak vypočtena hodnota, která udává, s jakou přesností si byli uživatelé jisti. Na základě limitů je pak rozřazen jako true, false nebo unknown. Obrázek musí splňovat přesnost alespoň 0.75 aby byl označen jako true a na druhou stranu maximálně hodnotu 0.25 aby byl označen jako false. Obrázky v rozmezí

těchto dvou hodnot jsou nejasné a označené jako unknown. Obě hodnoty jsou konfigurovatelné jako proměnné true_threshold a false_threshold.

```python
# Analýza výsledků pro jednotlivé datasety/složky
1 usage
def analyze_category_results(result_dict, done_folders):
    for folder_name, values in {key: value for key, value in result_dict.items() if key in done_folders}.items():
        # Inicializace
        true_list = []
        false_list = []
        unknown_list = []
        result_detail = {}
        # Dopočet ratia a rozřazení obrázků
        for image, tested_value in values['unknown_tested'].items():
            if image in values['unknown_true']:
                true_value = values['unknown_true'][image]
                ratio = true_value / tested_value
                result_detail[image] = ratio
                # Kontrola thresholdu
                if ratio >= true_threshold:
                    true_list.append(image)
                elif ratio <= false_threshold:
                    false_list.append(image)
                else:
                    unknown_list.append(image)
            else:
                result_detail[image] = 0
                false_list.append(image)

        # Volání dalších funkcí
        classify_and_move_images(folder_name, true_list, false_list)
        process_json_data(folder_name)
        # Ukládáme spočítanou míru ohodnocení
        save_results_to_file(result_detail, os.path.join(data_path_logs, f'{folder_name}_results_ratio.json'))
```

Obrázek 29 - Funkce analyze_category_results, zdroj: vlastní tvorba

Po rozdělení jednotlivých obrázků do tří kategorií je volána funkce classify_and_move_images která se stará o roztřídění obrázků do složek. Je vytvořena prázdná kopie složky předkládané při testech do prostoru 04_done. Každá taková složka obsahuje 3 podlsložky – true, false, unknown, kam jsou obrázky přesunuty dle výsledku. Z původního umístění, ze kterého byly předkládány uživatelům, jsou odstraněny.

```python
#Roztřídění obrázků do složek na základě rozdělení
1 usage
def classify_and_move_images(category, true_list, false_list):
    captcha_category_path = os.path.join(os.path.dirname(os.getcwd()), data_path_captcha, category)
    done_folder_path = os.path.join(data_path_done, category)
    # Příprava prázdných složek
    os.makedirs(done_folder_path)
    for subfolder in ['true', 'false', 'unknown']:
        subfolder_path = os.path.join(done_folder_path, subfolder)
        os.makedirs(subfolder_path)

    unknown_captcha_path = os.path.join(captcha_category_path, 'unknown')
    for file_name in os.listdir(unknown_captcha_path):
        file_path = os.path.join(unknown_captcha_path, file_name)

        #Rozdělení obrázků do složek
        if file_name in true_list:
            destination_folder = os.path.join(done_folder_path, 'true')
        elif file_name in false_list:
            destination_folder = os.path.join(done_folder_path, 'false')
        else:
            destination_folder = os.path.join(done_folder_path, 'unknown')

        shutil.move(file_path, destination_folder)

    # Přesun dat ze složky true
    true_captcha_path = os.path.join(captcha_category_path, 'true')
    for file_name in os.listdir(true_captcha_path):
        file_path = os.path.join(true_captcha_path, file_name)
        true_destination_path = os.path.join(done_folder_path, 'true')
        shutil.move(file_path, true_destination_path)
    # Snazání původní složky
    shutil.rmtree(captcha_category_path)
```

Obrázek 30 - Funkce classify_and_move_images, zdroj: vlastní tvorba

Posledním krokem je úprava souboru s odpověďmi uživatelů. Funkce process_json_data načítá soubor results.json a vybírá z něj záznamy, které se týkají složky ,která byla anotována a odebrána z aplikace CAPTCHA. Jsou uloženy do vlastního souboru a z tohoto results.json vymazána. Důvodem je případná další analýza odpovědí uživatelů a zajištění rozumné velikosti results.json. Společně se souborem obsahujícím výsledky je vytvořen soubor obsahující jednotlivé obrázky a jejich vypočítanou míru přesnosti.

```python
#Úprava dat v json logu
1 usage
def process_json_data(category):
    with open(os.path.join(data_path_logs, 'results.json'), 'r') as file:
        json_data = json.load(file)
    # Vybere záznamy pro danou složku/dataset
    category_records = [record for record in json_data['results'] if record.get('folder_name') == category]

    # Uložíme zvlášť
    save_results_to_file(category_records, os.path.join(data_path_logs, f'{category}_category_results.json'))

    # Smažeme z původního jsonu
    json_data['results'] = [record for record in json_data['results'] if record.get('folder_name') != category]

    # Uložíme původní json
    save_results_to_file(json_data, os.path.join(data_path_logs, 'results.json'))
```

Obrázek 31 - Funkce process_json_data, zdroj: vlastní tvorba

4.3.4 Pomocné soubory

Jak bylo již zmíněno, aplikace pro otestování potřebuje HTML stránku, která bude předkládána uživateli. Tato základní stránka je k nalezení ve složce projektu uvnitř složky templates.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CAPTCHA labeling</title>
    <! -- Navázání CSS -->
    <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
</head>
<body>
<! -- Vyskakovací zpráva v případě neúspěchu  -->
{% with messages = get_flashed_messages() %}
        {% if messages %}
            {% for message in messages %}
                <div class="flash">{{ message }}</div>
            {% endfor %}
        {% endif %}
    {% endwith %}
<form action = "/" method="POST">
    <! -- Prostor pro vygenerovanou CAPTCHA -->
    {{ captcha | safe }}
    <input type="hidden" id="selected_images" name="selected_images">
    <! -- Tlačítko pro odeslání -->
    <button type="submit">Submit</button>
</form>

<! -- Navázání Javascripotu -->
<script src="{{ url_for('static', filename='script.js') }}"></script>
</body>
</html>
</body>
</html>
```

Obrázek 32 - HTML template, zdroj: vlastní tvorba

V HTML jsou zmíněny dva další pomocné soubory. CSS soubor styles.css, zajišťující základní formátování.

```css
/* Rozložení CAPTCHA */
.captcha-container {
    display: grid;
    grid-template-columns: repeat(3, 100px);
    grid-auto-rows: 100px;
    grid-gap: 10px;
}

/* Pozicování obashu captcha */
.captcha-item {
    position: relative;
    overflow: hidden;
}

/* Nastavení formátu obrázku v CAPTCHA */
.captcha-item img {
    width: 100%;
    height: 100%;
    object-fit: contain;
    box-sizing: border-box;
    border: 2px solid transparent;
}

/* Zvýraznění vybraných obrázků po kliknutí */
.captcha-item img.selected {
    border-color: blue;

}
```

Obrázek 33 - CSS, zdroj: vlastní tvorba

Soubor s Javascriptem script.js umožňující uživateli ovládání CAPTCHA.

```javascript
document.addEventListener( type: "DOMContentLoaded", listener: function () : void {
    const selectedImages : any[] = [];

    // Funkce pro označení obrázků
    1+ usages
    function toggleImageSelection(imageElement) : void {
        const imageSrc : string = imageElement.getAttribute( qualifiedName: "src");

        if (selectedImages.includes(imageSrc)) {
            // Obrázek je již vybrán, odstraníme ho z výběru
            const index : number = selectedImages.indexOf(imageSrc);
            if (index > -1) {
                selectedImages.splice(index, deleteCount: 1);
            }
            imageElement.classList.remove( tokens: "selected");
        } else {
            // Obrázek není vybrán, přidáme ho do výběru
            selectedImages.push(imageSrc);
            imageElement.classList.add("selected");
        }

        // Aktualizujeme skryté pole s vybranými obrázky
        document.getElementById( elementId: "selected_images").value = JSON.stringify(selectedImages);
    }

    // Sledujeme kliknutí
    const imageElements : NodeListOf<Element> = document.querySelectorAll( selectors: ".captcha-item img");
    imageElements.forEach( callbackFn: function (imageElement : Element ) : void {
        imageElement.addEventListener( type: "click", listener: function () : void {
            toggleImageSelection(imageElement);
        });
    });

    // Sledujeme odeslání formuláře
    const form : HTMLFormElement = document.querySelector( selectors: "form");
    form.addEventListener( type: "submit", listener: function (event : SubmitEvent ) : void {
        // Zabraňujeme automatickému odeslání formuláře
        event.preventDefault();
        // Odeslání formuláře
        form.submit();
    });
```

Obrázek 34 - Javascript, zdroj: vlastní tvorba

4.3.5 config.py

Důležitým souborem je config.py nacházející se v kmenovém adresáři aplikace. Soubor obsahuje hodnoty, které je možné upravovat pro jinou míru přesnosti aplikace.

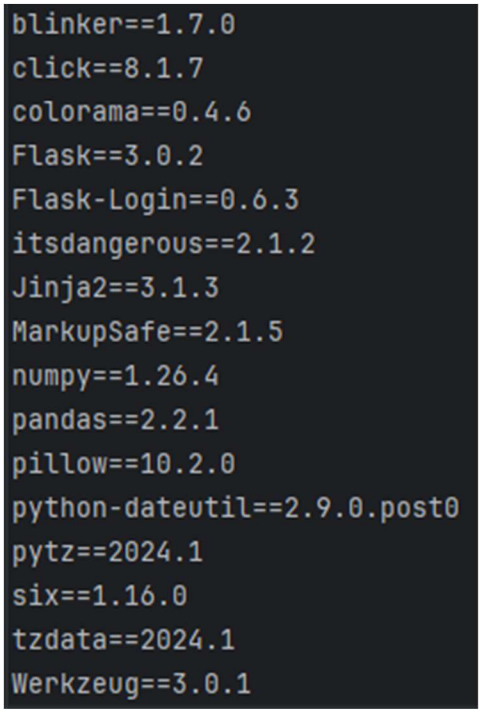
Název proměnné	Význam proměnné.
data_path	Cesta do úložiště souborů.
data_path_raw	Cesta do složky 01_raw, kde jsou uloženy nezpracované složky s obrázky k anotaci.

data_path_processed	Cesta do složky 02_processed, kde jsou uloženy připravené složky s obrázky k anotaci.
data_path_captcha	Cesta do úložiště aplikace, kde jsou uloženy složky s obrázky předkládané v CAPTCHA.
data_path_logs	Cesta do složky 03_logs, kde jsou uloženy json soubory s odpověďmi od uživatelů společně s výsledky.
data_path_logs	Cesta do složky 04_logs, kde jsou uloženy výsledné složky s roztříděnými obrázky.
required_true_pictures	Hodnota udávající požadovaný minimální počet obrázků v kontrolní složce true.
required_unknown_pictures	Hodnota udávající požadovaný minimální počet obrázků k anotaci.
image_size	Velikost obrázku udávaná pro přípravu dat.
secret_key	Konfigurační proměnná zajišťující session pro Flask.
captcha_true_min	Minimální počet výskytů true obrázku ve vygenerované CAPTCHA.
captcha_true_max	Maximální počet výskytů true obrázku ve vygenerované CAPTCHA.
anotation_threshold	Minimální počet předložení obrázku, aby byl ohodnocen jako anotovaný
true_threshold	Dolní limit přesnosti, při které je obrázek ohodnocen jako true
false_threshold	Horní limit přesnosti, při které je obrázek ohodnocen jako false

Tabulka 1 - Konfigurovatelné proměnné, zdroj: vlastní tvorba

4.3.6 requirements.txt

Soubor requirements.txt obsahuje všechny stahovatelné Python balíčky použité v aplikaci. Aby aplikace fungovala, musí být tyto balíčky stažené do Pythonu.



Obrázek 35 - Requierements.txt, zdroj: vlastní tvorba

4.4 Testování aplikace

Pro zjištění funkčnosti aplikace je vytvořen testovací scénář založený na anotaci satelitních snímků. Aplikace je implementována do procesu přihlášení do interních webových stránek vývojového týmu. Cílem testování je anotovat data na výskyt solárních panelů na střechách domů. K tomu je vybrána část malé obce v Jižních Čechách – Malovice.

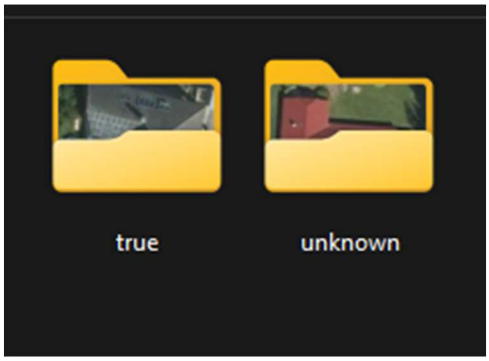
Obrázek 36 - Obrázek obce, zdroj: vlastní tvorba

Snímky jsou vytěženy ze satelitních snímku oblasti a snímek každého jednoho domu je uložen zvlášť jako vstup do aplikace. Domy jsou vytěženy na základě čísla domu. Některé objekty tak jsou větší než jiné. Celkem je získáno 98 obrázků domů.

4.4.1 Modul pro přípravu dat

Ve složce, na kterou odkazuje aplikace jsou vytvořeny složky nezbytné k běhu aplikace – 01_raw, 02_processed, 03_logs a 04_done.

Vytěžená data jsou uložena do složky „01_raw/solární panely“ neboť název složky reflektuje, jaká je v CAPTCHA položena otázka. Obrázky jsou uloženy do podsložky unknown a je pořízeno 10 obrázků s jasně identifikovatelnými solárními panely na střechách domů, které jsou uloženy do složky true.



Obrázek 37 - Připravené složky true a unknown, zdroj: vlastní tvorba

Nad takto připravenými daty je spuštěn script data_preparation.py, který kontroluje, zda je obrázků dostatečné množství dle konfiguračního souboru, zda existují potřebné podsložky a následně upravuje obrázky, aby odpovídali stanovené velikosti 100px na 100px.

Obrázek 38 - Obrázek po úpravě (vlevo) a před úpravou (vpravo), zdroj: vlastní tvorba

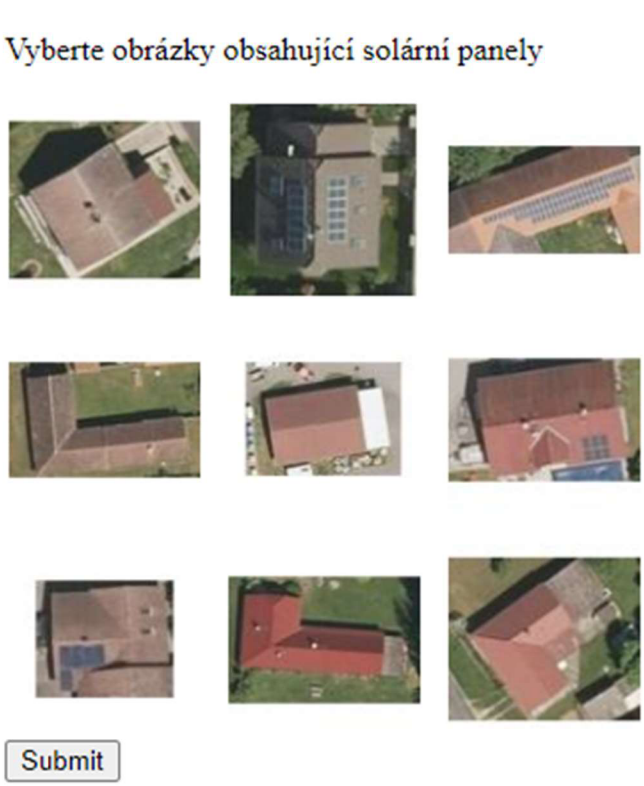
Obrázky jsou přejmenovány dle konvence. A připraveny na přesunutí do CAPTCHA.

Obrázek 39 - Přejmenované připravené obrázky, zdroj: vlastní tvorba

Zavolání funkce move_processed_folders_to_captcha jsou obrázky přesunuty a je možné nad nimi volat CAPTCHA.

4.4.2 Modul CAPTCHA

Pro otestování je modul spuštěn a je zkontrolována jeho funkčnost na základní testovací HTML stránce, před jeho nasazením. Dle nastavení obsahuje vždy minimálně 1 obrázek z potvrzených výskytů pro kontrolu.



Obrázek 40 - Vygenerovaná CAPTCHA, zdroj: vlastní tvorba

Uživatel vybírá obrázky dle zadání, v tomto případě obrázky obsahující solární panely. Vybrané obrázky jsou označeny, čímž dávají zpětnou vazbu uživateli. Pokud se uživatel zmýlí, může označený obrázek před odesláním znovu odznačit.

Obrázek 41 - CAPTCHA s označenými objekty, zdroj: vlastní tvorba

V případě úspěšného označení všech kontrolních obrázků je po odeslání a validaci vrácena zpráva o úspěšnosti. V testovacím případě vypadá následovně.

```
{
    "message": "Vybrány správné obrázky!",
    "status": "success"
}
```

Obrázek 42 - Zpětná vazba při splnění CAPTCHA, zdroj: vlastní tvorba

V případě neúspěšného vyplnění je vygenerována nová CAPTCHA a uživatel je upozorněn na chybu.

Obrázek 43 - CAPTCHA při nesprávném vyplnění, zdroj: vlastní tvorba

Aplikace je po základním otestování na testovací stránce nasazena po dobu jednoho týdne na interní přihlašovací webovou stránku vývojového týmu. Počet přístupů na stránku je obvykle kolem 20 denně.

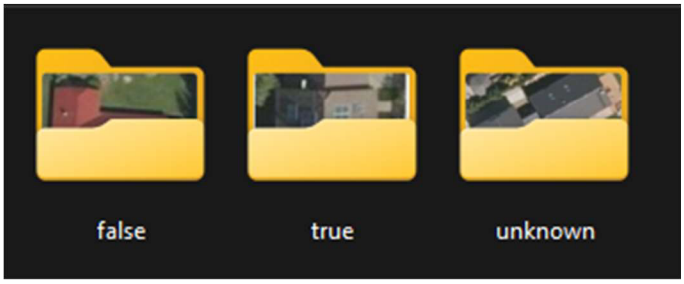
4.4.3 Modul pro vyhodnocení odpovědí

Po týdnu je aplikace stažena z přihlašovacího formuláře a je zhodnocen její běh. Během týdnu nejsou hlášeny žádné problémy ani výpadky CAPTCHA. Je spuštěn a otestován proces vyhodnocení odpovědí.

```
{'time': '2024-03-06 14:20:59',
'folder_name': 'solární panely',
'status': 'success',
'true_images': ['t1.jpg', 't0.jpg', 't4.jpg'],
'unknown_images': ['u0.jpg', 'u9.jpg', 'u8.jpg', 'u7.jpg', 'u3.jpg', 'u2.jpg'],
'selected_images': ['u0.jpg', 't0.jpg', 't4.jpg', 't1.jpg', 'u9.jpg']}
```

Obrázek 44 - Logovaná odpověď, zdroj: vlastní tvorba

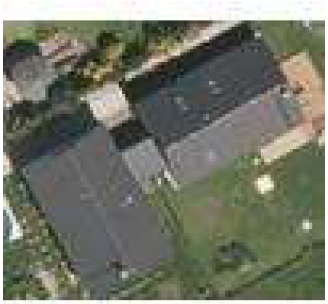
Vzhledem k tomu, že dataset splnil podmínku předložení nejméně předkládaného obrázku alespoň 10krát, může postoupit dále k vyhodnocení. Obrázky jsou na základě logovaných odpovědí z results.json rozděleny do 3. složek.



Obrázek 45 - Výsledné složky, zdroj: vlastní tvorba

Z původních 98 obrázků je vyhodnoceno jako true, tedy vyskytuje se na nich solární panel, 3 obrázky. Jako false, tedy panel neobsahují, je vyhodnoceno 91 domů. Zbylé 4 domy zůstávají ve složce unknown.

Všechny 4 domy mají tmavé střechy a na nich střešní okna. Je tedy pravděpodobné, že právě tato skutečnost zmátla uživatele.



Obrázek 46 - Matoucí obrázek, zdroj: vlastní tvorba

Detailní pohled na data nabízí jeden ze souborů json obsahující hodnotu vypočítanou vydělením pozitivního ohodnocení počtem předložení, na základě, které se data rozdělují do výsledných složek. Pokud se podíváme na jeden z domů ze složky unknown, je jeho přesnost ohodnocení 0,2842 což znamená, že je velmi blízko k nastavenému spodnímu limitu (0,25) a inklinuje více k možnosti, že se na domu solární panely nevyskytují. V konfiguračním souboru je možná úprava hranic pro false a true množinu. Vzhledem k tomu, že jsou ve složce unknown pouze 4 obrázky, není problém rozdělit je manuálně.

Po vyhodnocení výsledků aplikace je všech 98 obrázků anotováno ručně, čímž je zjištěno, s jakou přesností dokážou uživatelé pomocí aplikace obrázky anotovat. Pokud se podíváme na množinu true, dokázali uživatelé anotovat všechny domy se solárními panely.

5 Výsledky a diskuse

Aplikace je připravena pro implementaci na webovou stránku k ověřování uživatelů. Uživateli je předložen text CAPTCHA s alespoň jedním kontrolním obrázkem. Data potřebná pro funkčnost aplikace, primárně tedy obrázky, ze kterých se CAPTCHA generuje, jsou připraveny pomocí funkcí v scriptu data_preparation.py. Vyhodnocení provádí funkce ze scriptu data_evaluation.py

5.1 Modul pro přípravu dat

Aplikace využívá obrazová data a upravuje je do formátu vhodného k předložení uživateli. Data je třeba připravit a do aplikace dodat ideálně ve formátu 1 ku1. Název složky je předkládán v dotazu pro CAPTCHA.

5.1.1 Možné úpravy modulu pro přípravu dat

Možnou úpravou modulu pro přípravu dat je vytěžování dat z webové stránky či map. V aktuálním nastavení je třeba data shánět a do aplikace dodávat mimo aplikaci. V případě testování aplikace tak bylo nutné ručně vytěžit data z map, což bylo časově náročné. Pokud by se opakovaly požadavky na anotaci dat z map, bylo by vhodné vytvořit modul, který požadované obrázky automaticky vytěží.

Příprava dat je rychlá, avšak v aktuálním nastavení je limitována na obrázky formátované 1 ku 1.Některé obrázky, které tuto podmínku nesplňují mají po okrajích velká bílá pole, což snižuje viditelnost pro uživatele.

Využití názvu složky pro otázku se z počátku jevilo jako dobrý nápad, avšak není to nejvhodnější volba. Pokud bychom chtěli anotovat s úplně jinou otázkou, než je aktuální „Vyberte obrázky obsahující“ byla by nutná velká úprava aplikace. Navíc nelze používat více složek se stejnými daty. K tomu všemu existují určité znaky, které nelze do názvu složky zapsat, což je další omezující faktor.

5.2 Modul CAPTCHA

Aplikace využívá připravená data a předkládá je uživatelům ve formě mřížky o rozměru 3x3. Uživatel na základě otázky vybírá správné obrázky a svou odpověď odesílá stisknutím tlačítka. Odpověď uživatele je kontrolována na kontrolní množině a vyhodnocena. Zpětná vazba od uživatele je zapisována do souboru.

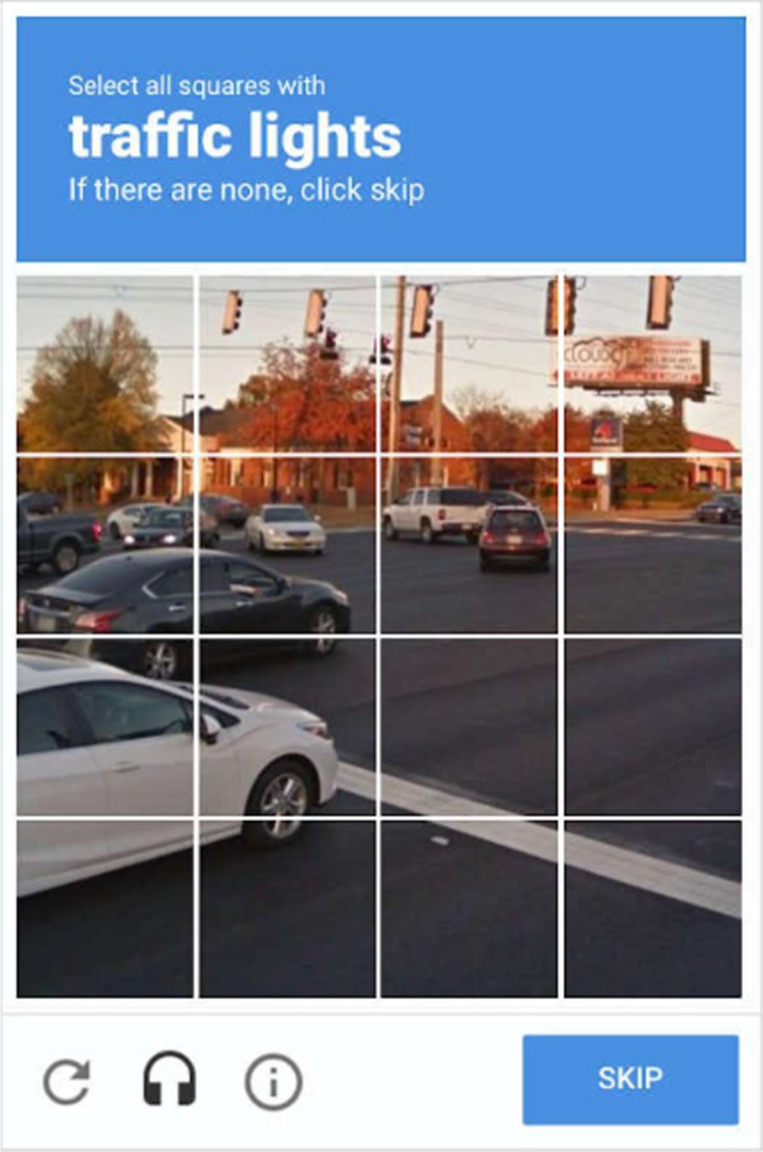
5.2.1 Možné úpravy modulu CAPTCHA

Aktuální aplikace podporuje pouze mřížky 3x3, což může být značně omezující. Jakákoliv změna v tomto ohledu vyžaduje velký zásah do kódu aplikace, neboť s mřížkou 3x3 počítají všechny pomocné funkce generující CAPTCHA i formátování v CSS.

Dalším problémem aplikace je přístupnost. Aplikace v aktuálním stavu vyžaduje po uživateli, aby neměl problémy se zrakem. Bylo by vhodné přidat alternativu pro zrakově postižené a jinak znevýhodněné osoby.

Aplikace nelimituje uživatele v odesílání CAPTCHA, z čehož lze usoudit, že i když bude náhodně klikat, jednou se trefí. Bylo by možné aplikaci upravit, sledovat jednotlivé uživatele a v případě pochybného chování mu přístup omezit.

Aplikace nedisponuje možností vygenerovat si jiné obrázky, pokud jsou ty aktuální nečitelné. Jedná se o obvyklou funkci ověření CAPTCHA a bylo by vhodné jej implementovat, neboť se nejedná o velký zásah do aplikace. Přidáním tlačítka, které by se navázalo na funkci generující CAPTCHA, by se nová CAPTCHA vygenerovala a předala uživateli.



Obrázek 47 - Obrázková CAPTCHA s ovládacími prvky(27)

Aplikace byla testována na stolním počítači a není zjištěno, jak se chová na mobilních zařízení a jestli je pro tato zařízení vhodná. Pro některá zařízení může být obrázek o velikosti 100px na 100px velký a špatně se sním bude manipulovat. Pozicování CAPTCHA se dá nastavit úpravou CSS, ale velikost obrázků je stanovena.

Obrázek 48 - CAPTCHA na mobilním zařízení, zdroj: vlastní tvorba

Vzhledem k tomu, jak je aplikace naprogramována a jak využívá kontrolní množinu, pouze true hodnot, bylo při testování odhaleno, že projde každý požadavek, pokud budou zaškrtnuta všechna políčka, nehledě na správnosti. Je tomu tak proto, že v případě zaškrtnutí všech polí jsou odhalena všechna testovací pole a aplikace uživatele pouští dále. Aby bylo

touto obejití aplikace zamezeno, bylo by možné přidat ke kontrolní množině true i kontrolní množinu false. Uživateli by byl předkládán náhodný počet obrázku ověřených pravdivých a ověřených nepravdivých společně s neověřenými a musel by odhalit obě kontrolní množiny.

Obrázek 49 - CAPTCHA při označený všech obrázků, zdroj: vlastní tvorba

Posledním návrhem na úpravu CAPTCHA je změna místa logování. V rámci testu nebyly odhaleny problémy s logováním do souboru, avšak při přístupu většího počtu uživatelů by se mohlo stát, že se soubor stane nedostupným a nepůjde do něj zapsat, což způsobí ztrátu dat nebo pád aplikace. Nabízí se logování do relační databáze, díky kterému by starost o vytížení souboru odpadla.

5.3 Modul pro vyhodnocení odpovědí

Modul pro vyhodnocení odpovědí zpracovává informace získané ze zpětné vazby od uživatelů. Na základě odpovědí rozděluje obrázky do 3 složek, a to true, false a unknown. Při vyhodnocení je dataset odebrán z CAPTCHA, neboť se bere jako anotovaný a nepotřebuje být nadále uživatelům předkládán

5.3.1 Možné úpravy modulu pro vyhodnocení odpovědí

I přes to, že máme údaje z volání CAPTCHA které skončilo nezdarem a nepuštěním uživatele, nejsou nijak využívány. Aktuálně neanalyzujeme chybné pokusy a přicházíme tím o informace. Je možné, že některý obrázek z kontrolní množiny není natolik jasný, jak si programátor myslel, a uživatelé ho často označují chybně.


```
{"time": "2024-03-05 08:24:53",
"folder_name": "solární panely",
"status": "error",
"true_images": ["t3.jpg", "t0.jpg", "t4.jpg"],
"unknown_images": ["u3.jpg", "u5.jpg", "u4.jpg", "u8.jpg", "u0.jpg", "u1.jpg"],
"selected_images": ["t3.jpg", "t0.jpg", "t4.jpg", "u0.jpg", "u4.jpg"]}
```


Obrázek 50 - Zpětná vazba při chybně vyplněné CAPTCHA, zdroj: vlastní tvorba

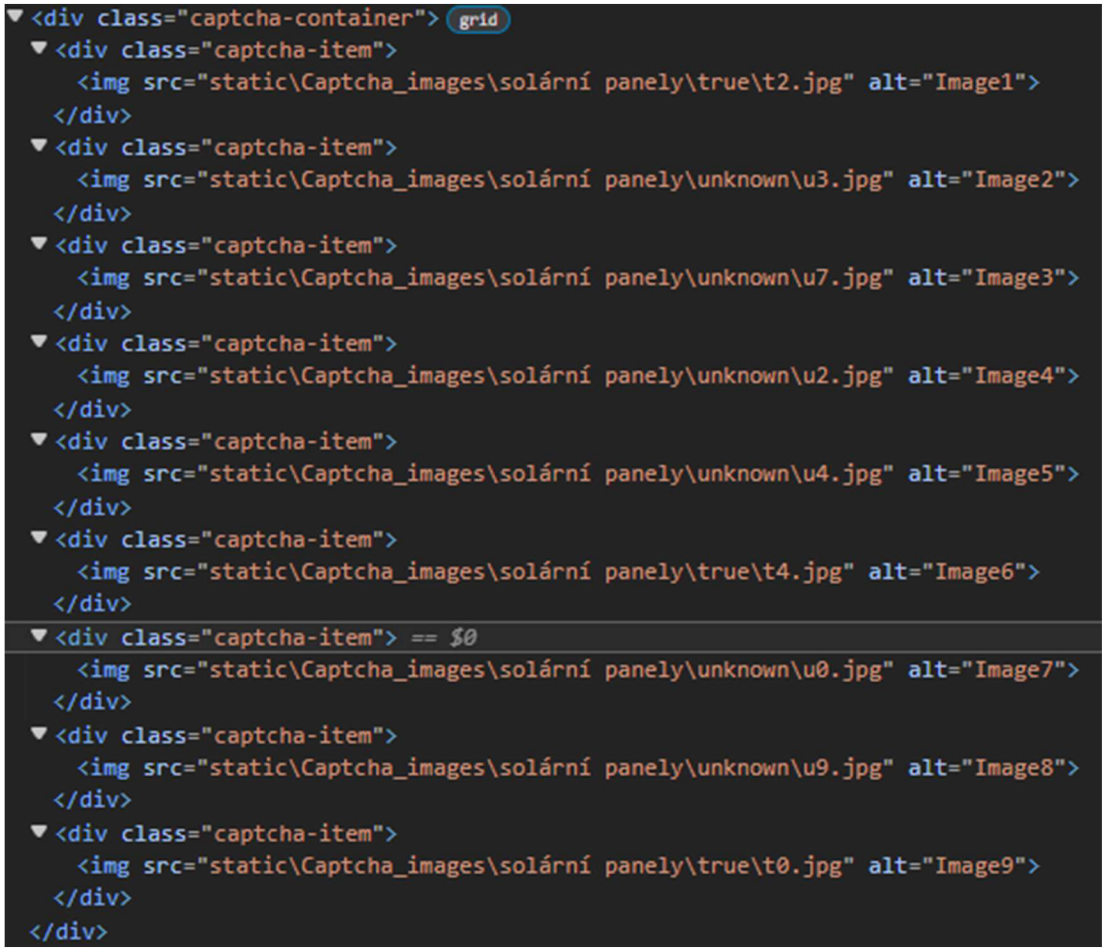
Aktuální modul pro vyhodnocení obrázky rozřazuje do tří složek na základě jednoho vypočtené hodnoty. Po rozdělení není tato hodnota přesnosti nijak promítnuta v daném obrázku.

5.4 Možné úpravy aplikace

Každá část aplikace funguje jako samostatná entita. Provázáním by bylo možné aplikace více automatizovat. Pro přidání nových dat a vyhodnocení stávajících je třeba spustit příslušné skripty. Pokud by se v nějakém časovém horizontu automaticky složka 01_raw procházela, mohla by data automaticky připravovat pro CAPTCHA. Stejně tak by bylo možné automaticky zjišťovat, jestli byl vybraný dataset už dostatečně krát předložen uživateli a pokud ano, byl by automaticky vyhodnocen. Bylo by však nutné zajistit nějaký defaultní dataset, který bude vždy k dispozici, neboť když budou všechny vyhodnoceny, nezbude v aplikaci žádný, který by mohl být předkládán, a aplikace nebude fungovat.

Aplikace je z velké části založena na náhodné volbě. Náhodně se vybírají datasety, ze kterých se náhodně vybírají předkládané obrázky. Rozložení tedy může být silně nerovnoměrné a v datasetech se mohou nacházet soubor, které byli předloženy vícekrát než jiné. I jeden obrázek tak může zdržet vyhodnocení celého datasetu.

Co se ochrany před boty týče, pojmenování obrázků je vidět ve zdrojovém kódu, což ubírá na bezpečnosti. Pokud by se názvy obrázků schovaly, nebo anonymizovaly, byla by ochrana vyšší.



Obrázek 51 - Vývojářská konzole na webu, zdroj: vlastní tvorba

6 Závěr

Práce byla zaměřena na problematiku ověřování uživatelů a pro tuto problematiku byla provedena analýza stávajících řešení. Metod ochrany a ověření uživatelů je na internetu mnoho. Hlavním cílem je ochránit choulostivá osobní data, neboť na internetu jsou přístupná všem, kdo má přístupové údaje. Bližší zaměření bylo na metodu CAPTCHA, jejíž hlavním cílem je odstínit boty od vyplňování formulářů. Nejznámější CAPTCHA od Google, ReCAPTCHA, svou funkcí inspirovala tuto práci. Starší verze využívaly odpovědi od uživatelů k anotaci dat. Díky ReCAPTCHA byl digitalizován celý archiv New York Times. Později byla předkládána i obrazová data z map a lidé se nevědomě podíleli na vylepšování aplikací společnosti Google. S rozvojem technologií bylo cíleno na eliminaci vyplňování testů, a tak je dnes využití CAPTCHA na internetové stránce těžko poznatelné. Nejnovější testy fungují na pozadí.

Cílem práce bylo napodobit starší verzi ReCAPTCHA a odpovědi od uživatelů používat k anotaci dat. Vize je taková, že by aplikace mohla být nasazena do malých vývojových týmů, případně do firem, které CAPTCHA využívají, a zároveň mají data, která je potřeba anotovat.

Na základě zjištěných poznatků byla navržena a naprogramována aplikace CAPTCHA, v jazyce Python, využívající framework Flask, která tak činí. Upravuje dodané obrázky a připravuje je pro generování CAPTCHA. Přípravou dat pro CAPTCHA byl splněn jeden z dílčích cílů této práce. Vygenerovaná CAPTCHA je vždy mřížka 9 obrázků (3 řádky, 3 sloupce) a úkolem uživatele je vybrat obrázky dle zadání. Zpětná vazba od uživatele je sbírána a využita pro anotaci dat. O anotaci se stará jeden z modulů aplikace. Data z datasetu jsou anotována až po splnění limitu předložených obrázků. Aby se nestalo, že uživatel náhodně vybere obrázky a CAPTCHA ho vždy pustí, obsahuje CAPTCHA kontrolní množinu obrázků s již ověřeným výskytem hledaného objektu.

Aplikace byla otestována na malém interním webu. Jako vstupní data byly použity satelitní snímky malé obce v jižních Čechách. Tím byl splněn i druhý dílčí cíl práce. Test proběhl úspěšně, anotované obrázky odpovídaly skutečnosti.

Na základě testu aplikace byly odhaleny nedokonalosti jednotlivých modulů aplikace a byly navrženy možné úpravy. Hlavními nedostatky aplikace jsou přístupnost a pouze jedna kontrolní množina. Vzhledem k tomu, že aplikace vyžaduje vybrání obrázků dle zadaného úkolu, nelze předpokládat, že by CAPTCHA mohla splnit zrakově postižená, nebo nějak

jinak zrakově znevýhodněná osoba. Co se pouze jedné kontrolní množiny týče, při testu bylo odhaleno, že projde každý požadavek, pokud budou zaškrtnuta všechna políčka, nehledě na správnosti. Případné řešení obou problémů je v práci navrženo.

Všechny cíle práce byly splněny. Aplikace byla navržena a naprogramována. Příprava dat pro nástroj CAPTCHA byla automatizována. Aplikace byla otestována na satelitních snímcích vesnice.

7 Seznam použitých zdrojů

1) CAPTCHA: The story behind those squiggly computer letters. BURLING, Stacey. *Phys* [online]. 2012 [cit. 2024-03-31]. Dostupné z: https://phys.org/news/2012-06-captcha-story-squiggly-letters.html

2) DASGUPTA, Dipankar, Arunava ROY a Abhijit NAG. *Advances in User Authentication*. 1. Springer International Publishing AG 2017, 2017. ISBN 978-3-319-58808-7.

3) 6 types of authentication for a comprehensive cybersecurity plan. JOHNSON, Anthony. *Connectwise* [online]. 2023, 09/06/2023 [cit. 2024-03-31]. Dostupné z: https://www.connectwise.com/blog/cybersecurity/types-of-authentication

4) BRODIĆ, Darko a Alessia AMELIO. *The CAPTCHA: Perspectives and Challenges*. 2020. Springer Cham, 2020. ISBN 978-3-030-29347-5.

5) NÚKIB v roce 2023 zaznamenal rekordní počet kybernetických incidentů. In: NÚKIB BRNO – MUČEDNICKÁ. *Národní úřad pro kybernetickou a informační bezpečnost* [online]. 2024 [cit. 2024-03-31]. Dostupné z: https://nukib.gov.cz/cs/infoservis/aktuality/2073-nukib-v-roce-2023-zaznamenal-rekordni-pocet-kybernetickych-incidentu/#:~:text=N%C3%A1rodn%C3%AD%20%C3%BA%C5%99ad%20pro%20kybernetickou%20a%20informa%C4%8Dn%C3%AD%20bezpe%C4%8Dnost%20%28N%C3%9AKIB%29,vlny%20DDoS%20%C3%BAtok%C5%AF%20veden%C3%A9%20zejm%C3%A9na%20prorusk%C3%BDmi%20hacktivistick%C3%BDmi%20skupinami

6) Use these 6 user authentication types to secure networks. JOHNSON, Kyle. *TechTarget* [online]. 18 Oct 2023n. l. [cit. 2024-03-31]. Dostupné z: https://www.techtarget.com/searchsecurity/tip/Use-these-6-user-authentication-types-to-secure-networks

7) Jak funguje párování čísel v nabízených oznámeních vícefaktorového ověřování pro Authenticator – zásady metod ověřování. In: MICROSOFT. *Microsoft Learn* [online]. 2023 [cit. 2024-03-31]. Dostupné z: https://learn.microsoft.com/cs-cz/entra/identity/authentication/how-to-mfa-number-match

Autentizace: biometrické metody. ČERMÁK, Miroslav. *Clever and smart* [online]. 8) 2013 [cit. 2024-03-31]. Dostupné z: https://www.cleverandsmart.cz/autentizace-biometricke-metody/#:~:text=Autentizace%3A%20biometrick%C3%A9%20metody%201%20Po%C5%99%C3%ADzen%C3%AD%20biometrick%C3%BDch%20charakteristik%20Aby,8%20Rozpozn%C3%A1v%C3%A1n%C3%AD%20obli%C4%8Deje%20%28face%20recognition%29%20...%20More%20items

Biometrické metody autentizace – současnost a perspektiva. LACKO, Ľuboslav. 9) *CIO* [online]. 2017 [cit. 2024-03-31]. Dostupné z: https://www.cio.cz/clanky/biometricke-metody-autentizace-soucasnost-a-perspektiva-1-3/

What is Biometrics? How is it used in security? AO KASPERSKY LAB. *Kaspersky* 1 [online]. 2024 [cit. 2024-03-31]. Dostupné z: https://www.kaspersky.com/resource-0) center/definitions/biometrics

Thumb Here, Please: Apple's iOS 8 Lets You Buy Things With a Fingerprint. In: 1 NBC UNIVERSAL. *NBC News* [online]. 2014 [cit. 2024-03-31]. Dostupné z: 1) https://www.nbcnews.com/tech/security/thumb-here-please-apples-ios-8-lets-you-buy-things-n206586

IT Explained: What Is SSO and What Does It Do? ISAK, Christopher. *Techacute* 1 [online]. 2022 [cit. 2024-03-31]. Dostupné z: https://techacute.com/what-is-sso/ 2)

Přihlásit se. RIOT. *Riot Games* [online]. 2024 [cit. 2024-03-31]. Dostupné z: 1 https://authenticate.riotgames.com/ 3)

Vaše digitální občanka Bankovní identita. BANK ID. *Bank iD* [online]. 2023 [cit. 1 2024-03-31]. Dostupné z: https://www.bankid.cz/#jak-funguje 4)

Přihlášení pomocí Identity občana do: Portál občana. In: DIGITÁLNÍ A 1 INFORMAČNÍ AGENTURA. *Identita občana* [online]. 2024 [cit. 2024-03-31]. 5) Dostupné z: https://nia.identitaobcana.cz/

What Is An Authentication Token? FORTINET, INC. *Fortinet* [online]. 2024 [cit. 2024-03-31]. Dostupné z: https://www.fortinet.com/resources/cyberglossary/authentication-token

SafeNet eToken PASS (dříve eToken 3000). In: ASKON INTERNATIONAL. *Askon* [online]. 2024 [cit. 2024-03-31]. Dostupné z: https://www.askon.cz/Produkty/Autentizace/OTP-autentizatory/SafeNet-eToken-PASS-drive-eToken-3000.html

Bezpečnost na internetu: Autentizační prvky a certifikáty. ČESKOSLOVENSKÁ OBCHODNÍ BANKA, A. S. *Průvodce podnikáním* [online]. 2020 [cit. 2024-03-31]. Dostupné z: https://www.pruvodcepodnikanim.cz/clanek/autentizacni-prvky-a-certifikaty/

Luis von Ahn. NATIONAL INVENTORS HALL OF FAME. *National Inventors Hall of Fame* [online]. 2023 [cit. 2024-03-31]. Dostupné z: https://www.invent.org/inductees/luis-von-ahn

Yahoo Mail Shows Captchas. In: CHITU, Alex. *Googlesystem* [online]. 2006 [cit. 2024-03-31]. Dostupné z: https://googlesystem.blogspot.com/2006/09/yahoo-mail-shows-captchas.html

Samples of text based CAPTCHAs. In: MOHINDER, Kumar. *Researchgate* [online]. 2022 [cit. 2024-03-31]. Dostupné z: https://www.researchgate.net/figure/Samples-of-text-based-CAPTCHAs_fig1_357859266

TESLA. Support: Full Self-Driving Computer Installations. TESLA. *Tesla* [online]. Palo Alto: Tesla, c2021 [cit. 2021-03-03]. Dostupné z: https://www.tesla.com/support/full-self-driving-computer?redirect=no

Why captchas are getting harder. In: VEGA, Edvard. *Vox* [online]. 2021 [cit. 2024-03-31]. Dostupné z: https://www.vox.com/22436832/captchas-getting-harder-ai-artificial-intelligence

Customizing the Look and Feel of reCAPTCHA. GOOGLE. *ReCAPTCHA help* [online]. 2016 [cit. 2024-03-31]. Dostupné z: https://developers.google.com/recaptcha/old/docs/customization

Implementing Google reCaptcha. JOG, Harshada. *Medium* [online]. 2021 [cit. 2024-03-31]. Dostupné z: https://medium.com/@harshadajog/implementing-google-recaptcha-d94a791ec836

Deciphering Old Texts, One Woozy, Curvy Word at a Time. GUGLIOTTA, Guy. *The New York Times* [online]. 2011 [cit. 2024-03-31]. Dostupné z: https://www.nytimes.com/2011/03/29/science/29recaptcha.html

Why CAPTCHA Pictures Are So Unbearably Depressing. THOMPSON, Clive. *Medium* [online]. 2021 [cit. 2024-03-31]. Dostupné z: https://onezero.medium.com/why-captcha-pictures-are-so-unbearably-depressing-20679b8cf84a

BURLING, Stacey. CAPTCHA: The story behind those squiggly computer letters. *Phys* [online]. 2012 [cit. 2024-03-31]. Dostupné z: https://phys.org/news/2012-06-captcha-story-squiggly-letters.html

Github. In: GITHUB. *Github* [online]. 2024 [cit. 2024-03-31]. Dostupné z: https://github.com/search?q=captcha+solver&type=repositories

Data Annotation in 2024: Why it matters & Top 8 Best Practices. KARATAS, Gulbahar. *AI Multiple research* [online]. 2024 [cit. 2024-03-31]. Dostupné z: https://research.aimultiple.com/data-annotation/#easy-footnote-bottom-3-29387

What is RLHF? BERGMANN, Dave. *IBM* [online]. 2023 [cit. 2024-03-31]. Dostupné z: https://www.ibm.com/topics/rlhf

SARKIS, Anthony. *Training Data for Machine Learning: Human Supervision from Annotation to Data Science*. 1. O'Reilly Media, 2023. ISBN 978-1492094524.

Text Annotations in the News Industry. In: POTTER, Rayan. *Data Science Central* [online]. 2021 [cit. 2024-03-31]. Dostupné z: https://www.datasciencecentral.com/text-annotations-in-the-news-industry/

10 Great Places to Find Free Datasets for Your Next Project. HILLIER, Will. *Career foundry* [online]. 2023 [cit. 2024-03-31]. Dostupné z: https://careerfoundry.com/en/blog/data-analytics/where-to-find-free-datasets/

Video annotation for machine learning: Opportunities and challenges.
3 SUPERANNOTATE. *SuperAnnotate* [online]. 2021 [cit. 2024-03-31]. Dostupné z:
5) https://www.superannotate.com/blog/video-annotation-for-machine-learning

8 Seznam obrázků a tabulek

8.1 Seznam obrázků

8.2 Seznam tabulek

Přílohy

Zdrojový kód aplikace v přílohách diplomové práce.