VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ROZPOZNÁVÁNÍ CAPTCHA

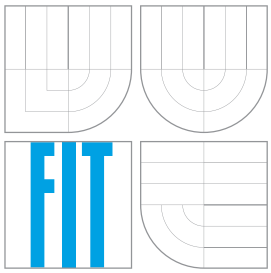
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAN KLIKA

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ROZPOZNÁVÁNÍ CAPTCHA

CAPTCHA RECOGNITION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAN KLIKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROSLAV ROZMAN, Ph.D.

BRNO 2014

Abstrakt

Tato práce se zabývá návrhem a implementací aplikace umožňující rozpoznávání CAPTCHA. Také se zabývá historií a vývojem CAPTCHA a způsoby jejího generování a možnými způsoby prolomení. Práce se zaměřuje na nové typy CAPTCHA, založené na obtížné segmentaci znaků. Hlavním cílem práce je tedy návrh a implementace nové segmentační metody, umožňující rozpoznání moderních CAPTCHA, konkrétně reCAPTCHA.

Abstract

This thesis describes the design and implementation of an application for breaking the CAPTCHA. It also describes the history and evolution of CAPTCHA and the ways of its generating and possible techniques of its breaking. This thesis focuses on the new types of CAPTCHA, based on hard character segmentation. So the main target of this thesis is the design and implementation of the new segmentation method, allowing the recognition of modern CAPTCHAs, especially reCAPTCHA.

Klíčová slova

CAPTCHA, strojové vidění, generování CAPTCHA, prolomení CAPTCHA, segmentace znaků, reCAPTCHA

Keywords

CAPTCHA, computer vision, CAPTCHA generating, CAPTCHA breaking, character segmentation, reCAPTCHA

Citace

Jan Klika: Rozpoznávání CAPTCHA, diplomová práce, Brno, FIT VUT v Brně, 2014

Rozpoznávání CAPTCHA

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rozmana, Ph.D.

.....
Jan Klika
27. května 2014

Poděkování

Rád bych poděkoval panu Ing. Jaroslavu Rozmanovi, Ph.D. za bezproblémovou spolupráci při tvorbě této práce.

Obsah

1 Úvod **3**

2 CAPTCHA **5**
2.1 Co je to CAPTCHA . . . 5
2.2 Typy CAPTCHA . . . 6
2.2.1 CAPTCHA proti necíleným útokům . . . 7
2.2.2 Standardizované CAPTCHA . . . 8
2.3 Historie prolomení CAPTCHA . . . 9
2.4 Možnosti využití . . . 10
2.4.1 Typické použití . . . 11
2.4.2 Vývoj umělé inteligence . . . 11
2.4.3 Digitalizace dokumentů . . . 12
2.4.4 Reklama . . . 12
2.5 Problémy . . . 13
2.6 Způsoby prolomení . . . 13
2.6.1 Levná pracovní síla . . . 13
2.6.2 Chyby v implementaci . . . 14
2.6.3 Vylepšené OCR . . . 14
2.6.4 Algoritmy simulující činnost mozku . . . 15
2.7 Proces generování . . . 15
2.7.1 Vývoj CAPTCHA . . . 16
2.7.2 Doporučení pro tvorbu bezpečné CAPTCHA . . . 18

3 Analýza problému **21**
3.1 Určení problému . . . 21
3.2 Analýza reCAPTCHA . . . 21
3.2.1 Návrh segmentační techniky . . . 22

4 Návrh aplikace **24**

5 Popis implementace **25**
5.1 Grafické uživatelské rozhranní . . . 25
5.2 Implementované metody . . . 27
5.2.1 Předzpracování . . . 27
5.2.2 Segmentace . . . 30
5.2.3 Post-segmentace . . . 31
5.2.4 Rozpoznání . . . 32

6 Testování **34**
6.1 reCAPTCHA 34
6.2 CAPTCHA API - Seznam.cz 35
6.3 Další schémata 36
6.4 Porovnání s jinými aplikacemi 38

7 Návrh pro zlepšení CAPTCHA **39**

8 Závěr **41**

A Obsah CD **44**

B Uživatelská příručka **45**

Kapitola 1

Úvod

S rozšířením internetu a množstvím poskytovaných služeb zdarma začaly vznikat systémy, jež této možnosti zneužívaly. Objevila se proto potřeba nějakým způsobem rozlišit skutečné osoby od počítačových programů. V reakci na to vznikla CAPTCHA[1], která měla sloužit jako test, který by byl pro člověka snadno vyřešitelný, ale pro počítačový program nikoli.

Nejznámějším a nejrozšířenějším typem CAPTCHA je opisování deformovaného textu z obrázku, existuje jich však mnoho druhů. Dalším velmi rozšířeným typem je CAPTCHA zvuková, která bývá často součástí textové verze jako alternativa pro osoby se zrakovým postižením. Tento způsob pracuje obdobně jako verze textová, avšak znaky jsou přehrávány s různým šumem v pozadí. Mezi další typy patří například obrazové, u kterých jde o rozpoznání předmětu na obrázku, nebo logické, jež jsou založeny na jednoduché logické úloze, např. "1+1=". Úspěšnost těchto typů je však založena na jejich malém rozšíření, a tím malé motivaci vytvářet programy, jež by si s těmito typy poradily.

Hlavním cílem této práce bude navrhnout a implementovat způsob strojového rozpoznání CAPTCHA. Práce bude zaměřená na prolomení reCAPTCHA, tedy implementace CAPTCHA vlastněnou společností Google. Nalezené metody rozpoznání pak budou aplikovány i na další typy CAPTCHA.

V následující kapitole této práci se podíváme na důvody, proč CAPTCHA vznikla, a na její vývoj. Uvedeme si různé typy CAPTCHA a seznámíme se pracemi zaměřenými na její prolomení. Dále se podíváme na její typické využití i její další, ne tak zřejmé, přínosy a problémy spojené s jejím užíváním. Uvedeme si různé způsoby obejití ochrany pomocí CAPTCHA, z nichž jeden se bude týkat samotného prolomení CAPTCHA, tedy automatického rozpoznaní znaků, které bude cílem této práce. Seznámíme se také s procesem jejího generování a uvedeme různá doporučení k vytvoření bezpečné CAPTCHA, známá z dřívějších prací.

Ve třetí kapitole se seznámíme s problémem, který v této práci budeme řešit, tedy prolomení reCAPTCHA, a dále také s jeho analýzou a návrhem jeho dalšího řešení.

V kapitole čtvrté se krátce seznámíme s návrhem aplikace, jež bude v rámci této práce vytvořena.

V následující kapitole si uvedeme detailnější informace týkající se implementace vytvořené aplikace. Půjde zejména o popis grafického uživatelského rozhraní a popis implementovaných metod pro zpracování CAPTCHA.

V šesté kapitole si uvedeme způsob a nejlepší dosažené výsledky při rozpoznání vybra-

[1] CAPTCHA je akronym pro „Completely Automated Public Turing test to tell Computers and Humans Apart“. Jde tedy o automatický veřejný Turingův test [17] k odlišení počítačů a lidí.

ných dvou typů CAPTCHA, uvedeme si také možnosti rozpoznání dalších typů CAPTCHA a porozvnáme dosažené výsledky s jinými aplikacemi.

V kapitole sedmé pak uvedeme doporučení pro tvorbu bezpečné CAPTCHA založené na získaných poznatcích.

Poslední kapitola pak obsahuje shrnutí dosažených výsledků a návrh možných vylepšení.

Kapitola 2

CAPTCHA

V této kapitole se podrobněji seznámíme s tím, co to vlastně CAPTCHA je, jak vznikla a k čemu všemu nám slouží. Uvedeme si také její nevýhody a způsoby jejího obcházení.

2.1 Co je to CAPTCHA

CAPTCHA je akronym pro „Completely Automated Public Turing test to tell Computers and Humans Apart“. Jde tedy o automatický veřejný Turingův test [17] k odlišení počítačů a lidí. V původním Turingově testu kladl testující otázky dvěma hráčům, přičemž jeden z nich byl člověk a jeden stroj. Oba však předstírali, že jsou člověk. Úkolem testujícího bylo pouze na základě jejich odpovědí rozhodnout, kdo z nich je člověk a kdo stroj. CAPTCHA je v tomto směru podobná, jen testujícím není člověk, ale počítač. CAPTCHA tedy není pouze zdeformovaný text v obrázku, je to obecně jakýkoliv automaticky generovaný test, který může většina lidí splnit, ale současné počítačové programy ne.

Jak již bylo zmíněno v úvodu, potřeba vytvořit mechanizmus povolující přístup k určitým zdrojům nebo službám pouze lidským uživatelům vznikala s narůstajícím počtem počítačových programů, agentů, které služeb zneužívaly. Mezi první takto postižené služby patřily webové vyhledavače a cenové srovnavače. [11] Jejich služby mohly být poskytovány uživatelům zdarma výměnou za jejich loajalitu či zobrazování reklam. V momentě, kdy byly jejich výsledky dále zpracovávány pomocí agentů, přestaly mít tyto společnosti přímý kontakt s uživateli.

První zmínka o nápadech spojených s automatickými Turongovými testy se pravděpodobě objevila v nepublikovaném rukopisu Moni Naor z roku 1996 [11]. Tato práce obsahuje některé klíčové pojmy a poznání, neobsahuje však žádné návrhy pro automatické Turingovy testy ani formální definice. Původní návrh CAPTCHA vychází z kryptografie, konkrétně z procesu identifikace. Ta je založena na tom, že strana A, která chce prokázat identitu straně B, prokáže schopnost efektivně spočítat funkci s klíčem, kterou jiný uživatel, který klíč nemá, spočítat nedokáže. Proces identifikace se tedy skládá z výzvy zadané stranou B a odpovědi spočtené stranou A. Pro získání mechanizmu pro rozlišení osob od agentů navrhuje místo výpočetní funkce s klíčem použít takový úkol, ve kterém člověk vyniká, zatím co stroje jej nedokážou vyřešit v rozumném čase. Vlastnosti, které by tyto úkoly měly mít:

- Musí být lehce generovatelné velké množství daného problému.
- Lidé musí být schopni problém vyřešit s velmi malým množstvím chyb. Způsob odpovědi musí být jednoduchý.

- Nejlepší známé programy pro řešení těchto problémů musí neuspět na nezanedbatelné části problémů, i když bude mechanizmus generování problému znám.
- Prezentace problému musí být nenáročná.

Jako možné zdroje problémů pro Turingovy testy jsou v práci navrhovány například rozpoznávání pohlaví, rozpoznávání výrazu ve tváři, nalezení částí těla, rozpoznávání tvarů, ručně psaného písma, doplňování slov do vět, rozpoznávání řeči a další. [11] Z těchto uvedených zdrojů problemů se v současné době používá nejvíce rozpoznávání písma, řeči a v ojedinělých případech také rozpoznávání tvarů.

První praktický příklad automatických Turingových testů byl systém vyvinutý společností Altavista [8] k prevenci před automatickou registrací webových stránek "boty". Tento systém byl založen na obtížnosti čtení slabě zdeformovaných znaků. Ve své době fungoval celkem dobře, ale kladl si za cíl pouze porazit současnou implementaci optického rozpoznávání znaků (Optical Character Recognition, ORC). Samotný pojem CAPTCHA, spolu s několika praktickými návrhy pro automatické Turingovy testy, byl představen v roce 2000 [1].

2.2 Typy CAPTCHA

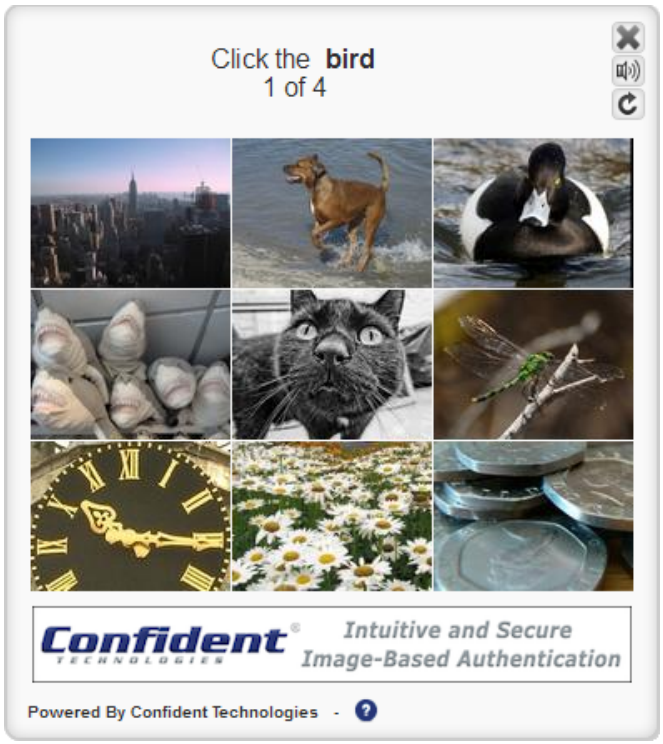
Jednotlivých typů CAPTCHA bylo navrženo více. Těmi nejpoužívanějšími v současné době jsou

- textové CAPTCHA - Je o opisování znaků z obrázku, přičemž znaky jsou různě deformované a umístěné tak, aby byly hůře strojově rozpoznatelné. Celkově jde o nejpoužívanější typ CAPTCHA v současné době. Příkladem může být CAPTCHA používaná na Wikipedii (Obr. 2.1)

Obrázek 2.1: Textová CAPTCHA používaná na Wikipedii, zdroj wikipedia.org

- zvukové CAPTCHA - Jde o rozpoznání hlásek přehraných ve zvukové sekvenci. Jednotlivé hlásky jsou vyslovovány různými mlůvčími a v pozadí se vyskytuje značný šum pro zabránění automatického rozpoznání analyzátory hlasu.
- obrázkové CAPTCHA - Jedná se o jednoduché úkoly nad sadou obrázků. Často jde o rozpoznávání druhů zvířat (viz obr. 2.2) či výběr nehodících se obrázků ze skupiny obrázků s podobnou tematikou.

CAPTCHA je také možné dělit z pohledu typu útoku, proti kterému mají chránit. Jedná se o ochranu před útoky necílenými a cílenými.



Obrázek 2.2: Obrázková CAPTCHA, zdroj http://confidenttechnologies.com

Necílené utoky provádějí často jednoduší boti[1], kteří prohledávají webové stránky, a při nalezení formulářů se přes ně snaží vkládat reklamní sdělení. Cílem jsou tedy nechráněné formuláře pro vkládání příspěvků do internetových diskuzí, kde je vložený příspěvek ihned viditelný ostatním návštěvníkům webové stránky. Jedná se o necílený útok, cílem autora bota tedy není přidat reklamní sdělení na konkrétní webovou stránku, ale vložit jej na co nejvíce takových, kde je to možné. Také servery, na které se tito boti zaměřují, nepatří velkým společnostem s vysokou návštěvností, jedná se převážně o soukromé webové prezentace, blogy či webové stránky malých společností a organizací.

2.2.1 CAPTCHA proti necíleným útokům

Díky tomu, že útok není nijak upraven pro konkrétní webovou stránku a přítomnost CAPTCHA se nepředpokládá, je možné využít takový typ CAPTCHA, který návštěvníky webové stránky nijak neobtěžuje. Ta je založena na obvyklých vlastnostech těchto botů a jejich chování. Jelikož tito boti automaticky vyhledávají a odesílají automaticky vyplněné formuláře, nezabývají se interpretací JavaScriptového kódu ani vzhledem webové stránky definovaným v CSS. Jednoduchým řešením jak boty odhalit tedy může být vytvoření skrytého formulářového prvku, který bude následně při zpracování formuláře zkontrolován, zda je prázdný. Jelikož boti neviditelnost prvků často neřeší a vyplňují všechna pole formuláře, je bota snadné odhalit. Jiným způsobem může být opět vytvoření formulářového prvku, tentokrát s popiskem např. "5+3=", a jeho skrytím a nastavením správné hodnoty pomocí JavaScriptu. Bot pravděpodobně vyplní pole špatnou hodnotou, zatím co za uživatele je výsledek doplněn automaticky. V případě, že by návštěvník stránky měl zakázaný Ja-

[1] počítačový program, který autonomně nebo na pokyn obsluhy opakovaně vykonává menší úkoly [23]

vaScript, pak jednoduchý výpočet snadno vyplní do zobrazeného pole. Dalším jednoduchým příkladem může být zobrazení stránky s potvrzením po odeslání formuláře. Pro většinu jednoduchých botů končí úkol odesláním právě prvního formuláře a jakési potvrzení nebudou brát na vědomí.

Další možností je vytvoření vlastní, např. textové, CAPTCHA. Ta by proti sofistikovanému cílenému útoku pravděpodobně neuspěla, ale pro boty prohledávající různé webové stránky bude neznámá a je velká šance, že si s ní neporadí. Pro necílené útoky lze navíc použít takové techniky, které úplně zabrání rozpoznání daného textu, který je však pro návštěvníka stránky velmi snadno čitelný. Příkladem může být například rozdělení obrázku dvě na části umístěné v různých částech dokumentu a jejich následné spojení pomocí správného pozicování pomocí CSS. (viz obr. 2.3) Jelikož ani jeden z obrázků neobsahuje celé znaky, není bot schopen znaky přečíst pouze na základě jednoho z nich. Pro správné rozpoznání CAPTCHA by tedy bylo nutné zjistit, které dva obrázky jsou požadované části CAPTCHA, ty spojit a následně provést rozpoznání znaků. Pro necílený útok je tento způsob naprosto účinný a pro uživatele snadno čitelný. Cílený útok by si s uvedenou technikou však snadno poradil. Hlavním úkolem by bylo nalezení dvou obrázků, které po vykreslení leží vedle sebe, a to i přes jejich různé umístění v dokumentu a jejich identifikátor či třídu v CSS.

Obrázek 2.3: Rozdělení obrázku na části a pozicování pomocí CSS

2.2.2 Standardizované CAPTCHA

Dalším řešením pak může být využití standardizovaných CAPTCHA - tedy již existujících CAPTCHA poskytovaných jako služby velkými společnostmi. Tyto služby mohou být bezplatné (např. reCAPTCHA či Seznam CAPTCHA API), nebo také placené (např. captcha.com). Jejich velkou výhodou je jejich snadné nasazení, kdy je společnostmi poskytováno jednoduché a bezpečné rozhraní pro použití ve webových stránkách. Vzhledem k vysoké frekvenci výskytu těchto CAPTCHA na webových stránkách jsou častým terčem cílených útoků. Na jednu stranu je možné je díky tomu považovat za odolnější proti cíleným útokům, na druhou stranu však nastává problém v situaci, kdy je daný typ CAPTCHA překonán, a tím se tisíce webových stránek stávají v tomto směru nezabezpečenými, dokud není poskytnuta nová verze CAPTCHA. Tyto CAPTCHA již nemohou využívat neznalosti nějaké vlastnosti útočníky, musí se spoléhat čistě na konkrétní problém, který je pro počítačový program nevyřešitelný. U textových CAPTCHA to je například rozmístění znaků tak, aby byla nemožná jejich segmentace.

Cíleným útokům tedy musí vzdorovat převážně velké společnosti, které nabízejí služby s větší hodnotou pro útočníky než je úsilí pro samotný útok. Nejčastějším cílem útoků je registrace e-mailových účtů u známých společností (Google, Yahoo, Microsoft), které jsou považovány za bezpečnější a e-mailové adresy z těchto domén jsou málokdy blokovány jako spam. Dalším častým terčem je registrace u společností nabízejících webový prostor zdarma, který je následně útočníky využíván pro šíření nelegálního obsahu.

Obrázek 2.4: CAPTCHA API od Seznam.cz

Obrázek 2.5: CAPTCHA společnosti Google

Obrázek 2.6: CAPTCHA používaná na Ulozto.cz

Obrázek 2.7: Ukázky CAPTCHA z captcha.com

2.3 Historie prolomení CAPTCHA

V této sekci si uvedeme některé z prací, které se snažily o prolomení CAPTCHA. Již z principu panuje neustálý boj mezi těmi, kteří CAPTCHA vytvářejí, a těmi, kteří se je snaží prolomit. Každé prolomení CAPTCHA tedy většinou vede k jeho vylepšení. V roce 2003 Mori a Malik [9] využili algoritmu pro rozpoznávání složitých objektů pro prolomení Gimpy (využívající interference znaků s šumem, viz obr. 2.8) a EZ-Gimpy (využívající texturované pozadí) s úspěšností 33% a 92%. Moy a kol. [10] vyvinuli techniku odhadu deformace s 99% úspěšností na EZ-Gimpy a 78% úspěšností na Gimpy-r se čtyřmi znaky.

V roce 2005 Chellapilla a kol. úspěšně prolomil řadu CAPTCHA s úspěšností 4,89% - 66,2%.[15]. Dřívější útoky zahrnuje také projekt PWNtcha [13]

Obrázek 2.8: GIMPY CAPTCHA

V roce 2006 Jeff Yan a El Ahmad [25] prolomili většinu vizuálních schémat poskytovaných serverem Captchaservice.com, což byla veřejně dostupná webová služba pro generování CAPTCHA, s úspěšností téměř 100% pouze na základě počtu pixelů v každém segmentovaném znaku, přestože tato schémata byla odolná vůči nejlepším OCR softwarům na trhu. V roce 2008 poté vyvinuli nové techniky pro segmentaci znaků [26] pro útok na množství CAPTCHA, včetně těch používaných společnostmi Microsoft, Yahoo! a Google. Úspěšnost segmentace například proti CAPTCHA od firmy Microsoft byla 92%. V roce 2010 Yanův tým prolomil textovou CAPTCHA [5], částečně založenou na principu vnímání tvarů (obr. 2.9), pomocí spojování černých a sdílených bílých komponent do jednotlivých znaků.

Obrázek 2.9: CAPTCHA založená na vnímání tvarů

V roce 2011 Burszstein a kol. ukázal, že 13 CAPTCHA na oblíbených webových stránkách byly zranitelné proti automatickým útokům [2], které ale nebyly úspěšné na složitější schémata jako reCAPTCHA a vlastní schéma společnosti Google. Ten samý rok Yanův tým zveřejnil efektivní útok na obě tato schémata [6]. Xu a kol. v roce 2012 [24] prolomil CAPTCHA využívající pohyblivých obrázků v NuCaptcha, která zobrazovala jezdící text se zvýrazněnými znaky k opsání.

2.4 Možnosti využití

Původní a nejčastější oblastí využití CAPTCHA je bezpečnost. Zde slouží k účelu, pro který byla navrhována, tedy rozlišování mezi lidskými uživateli a programy. Uvedeme si některé z jejích nejčastějších aplikací.

2.4.1 Typické použití

Online hlasování – V listopadu 1999 spustil server slashdot.com online hlasování o tom, která je nejlepší vysoká škola počítačových věd [1]. Přiznejme si, že to byla nebezpečná otázka pro online hlasování. Pro zabránění opakovaného hlasování se zaznamenávaly IP adresy hlasujících, nicméně studenti Carnegie Mellon našli způsob jak hlasovat opakovaně a vytvořili program, který jejich univerzitě přidával tisíce hlasů. Skóre univerzity začalo rapidně stoupat, načež další den zareagovali studenti MIT a přišli s vlastním hlasovacím programem. Hlasování nakonec skončilo s výsledkem 21 156 hlasů pro MIT a 21 032 pro Carnegie Mellon, ostatní školy přitom neměly ani 1 000 hlasů. Dalo by se tedy výsledkům online hlasování bez použití CAPTCHA věřit?

Bezplatné e-mailové služby – Mnoho společností nabízí bezplatné e-mailové služby, téměř všechny se však potýkají s problémem, kdy počítačové programy, tzv. "boti", registrují tisíce e-mailových účtů každou minutou. Tím nejen zabírají volné e-mailové adresy klientům, ale také využívají získané účty k následnému rozesílání spamu, tedy nevyžádané pošty. Také zde je řešením CAPTCHA, díky které se omezí možnost registrovat nový účet pouze pro skutečného uživatele.

Vyhledávací roboty – Některé webové stránky nechtějí být indexovány vyhledávacími stroji. Mohou použít html tag pro prevenci čtení jejich obsahu vyhledávacími roboty, ten však pouze oznamuje, že si autor stránky nepřeje, aby byla indexována. Pro skutečné zabránění vyhledávacím robotům ve vstupu na stránku je opět možné použít CAPTCHA [1].

Spam – CAPTCHA také nabíží rozumné řešení pro ochranu před e-mailovým spamem [19]. Řešením je přijmout e-mail pouze v případě, že víme, že odesílatelem je skutečný člověk. Některé společnosti, jako například www.spamarrest.com, nabízejí takovéto řešení. Jejich systém automaticky blokuje přijaté zprávy a zasílá jim automatickou odpověď s odkazem na CAPTCHA, teprve po jejím správném vyřešení je e-mail doručen uživateli.

Obrana před slovníkovými útoky – Dalším častým využitím CAPTCHA je zabránění ve slovníkovém útoku na prolomení hesla [19], kde se například po třetím špatně zadaném heslu musí s každým dalším pokusem vyřešit také CAPTCHA, čímž se zabrání programu v iteraci přes všechna možná hesla.

V dnešní době se s CAPTCHA setkáváme velmi často. Dá se říci, že CAPTCHA nalezneme téměř všude tam, kde je potřeba zabánit automatickému využívání služby poskytované webovou stránkou.

2.4.2 Vývoj umělé inteligence

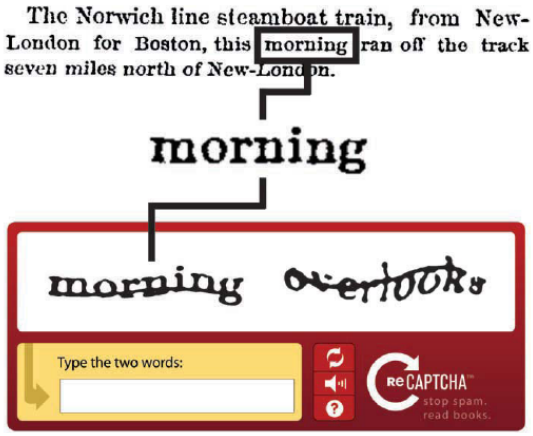
Mezi ne tak zřejmé přínosy CAPTCHA patří také rozvoj umělé inteligence. CAPTCHA je kryptografický protokol, jehož předpoklad složitosti je založen na problému umělé inteligence. [19]

Důležitá část úspěchu moderní kryptografie je založena na přesném a jasném definování předpokladu, díky kterému považujeme kryptografický protokol za bezpečný. To umožňuje vyhodnocení těchto předpokladů a pokusy o jeho prolomení.

Precizním stanovením nevyřešeného problému umělé inteligence můžeme urychlit vývoj umělé inteligence: většina problémů umělé inteligence, jež byly přesně definovány a zveřejněny, byly následně vyřešeny. Příkladem mohou být šachy. Z tohoto důvodu dává smysl použití pro bezpečnostní účely takových problémů umělé inteligence, které jsou také užitečné. Pokud je tedy problém umělé inteligence, na němž je úkol v CAPTCHA založen, užitečný, je CAPTCHA tzv. win-win situací: buď není CAPTCHA prolomena, čímž získáváme nástroj k rozlišení lidí od strojů, nebo CAPTCHA prolomena je, ale došlo k vyřešení užitečného problému umělé inteligence.

2.4.3 Digitalizace dokumentů

S rozmachem používání CAPTCHA, kdy lidé po 10s intervalech trávili stovky hodin denně opisováním textu z obrázku, vyvinuli na univerzitě Carnegie Mellon systém reCAPTCHA [1], který měl sloužit k něčemu víc, než jen k rozeznávání lidí a počítačových programů. Rozhodli se využít výkonu lidského mozku používaného při přepisu slov v CAPTCHA k digitalizaci knih a časopisů (obr. 2.10). Naskenovaný text se podrobil analýze dvou OCR programů a v případě, že se neshodly, bylo toto slovo převedeno do CAPTCHA a doplněno slovem již známým. Předpokládá se, že je-li kontrolní slovo správně, je správně také slovo nejasné. Po shodné interpretaci rozpoznávaného slova dvěma lidmi je i toto považováno za známé. Tento systém se stal velmi populárním. V roce 2009 jej koupila společnost Google a v současné době je denně reCAPTCHA vyřešena více než 200 miliony uživateli [1]. Kvůli značné deformaci textu jako ochraně proti prolomení se stala reCAPTCHA obtížně řešitelná i pro běžné uživatele. Její schéma bylo tedy změněno a v současné době je úkolem opsat pro člověka snadno čitelné číslice doplněné o obrázek s číslem domu, čímž se pomáhá automatickému zjišťování čísel domů pro mapy společnosti Google na základě snímků z Google Street View.

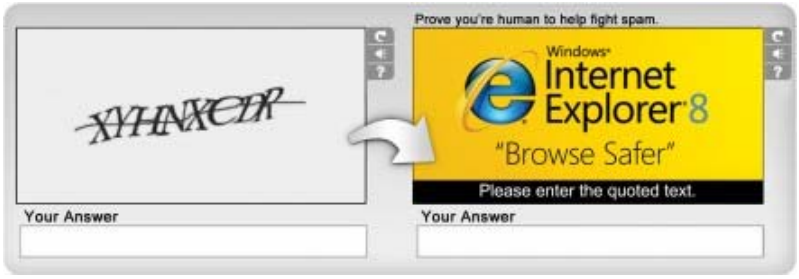


Obrázek 2.10: reCAPTCHA, zdroj[1]

2.4.4 Reklama

Vzhledem k počtu lidí, kteří CAPTCHA řeší každý den, se objevily společnosti, jež se tohoto potenciálu snaží využít a použít CAPTCHA jako reklamní kanál [16]. Nahrává jim také to, že psaní značně přispívá k memorování. Příkladem může být společnost SolveMedia, která nabízí placenou reklamu prostřednictvím CAPTCHA. Uživatelé tak místo deformovaného

textu opisují reklamní slogany. Otázkou však zůstává, jeslti se takový systém vůbec může nazývat CAPTCHA, jelikož text v reklamních sděleních je velmi snadno čitelný a množství zobrazovaných obrázků je omezeno počtem přeplatitelů služby.



Obrázek 2.11: CAPTCHA s reklamou, zdroj[16]

2.5 Problémy

Většina implementací CAPTCHA je založena na čtení textu nebo jiných vizuálních úkolech. To zabraňuje v užívání webu lidem slepým nebo s jiným zrakovým postižením. Příkladem mohou být barvoslepí lidé, pro něž jsou úkoly založené na použití barev neřešitelné. CAPTCHA však nemusí být vždy vizuální. Lze použít jakýkoliv složitý problém umělé inteligence, jako například rozpoznávání řeči. To bývá součástí řady implementací jako alternativa pro osoby se zrakovým postižením. Audio CAPTCHA je však použitelná pouze jako alternativa pro CAPTCHA textové, její použití u obrázkových typů je nemožné. Dalším problémem je také nemalé procento lidí se zrakovým i sluchovým postižením, pro které jsou stránky používající CAPTCHA bez pomoci jiné osoby nepřístupné. Jen ve Velké Británii bylo v roce 2010 podle [14] 250 000 hluchoněmých.

Dalším problémem je také zobrazování CAPTCHA na různých zařízeních. Současná nová zařízení samozřejmě nemají se zobrazováním CAPTCHA na svých displejích žádný problém, ale například na starších mobilních telefonech s nižším rozlišením displeje může být zobrazení CAPTCHA nemožné a některé stránky či služby se mohou pro tuto skupinu uživatelů stát nedostupnou.

2.6 Způsoby prolomení

Již od vzniku CAPTCHA se samozřejmě začaly objevovat pokusy o její prolomení, které následně vedly k jejímu vylepšení, a tím k vytvoření další výzvy pro její prolomení. Obecně existují 3 přístupy, jak zabezpečení pomocí CAPTCHA překonat. S těmi se seznámíme podrobněji a zmíníme i čtvrtý přístup, o kterém však v současné době není k dispozici mnoho informací.

2.6.1 Levná pracovní síla

Prvním přístupem je využití levné pracovní síly nebo nic netušících uživatelů. První možnost, tedy využití levné pracovní síly, není řešení nejvýkonnější ani nejlevnější, ale je daleko nejúčinnější. Celá myšlenka je založena na využití levné pracovní síly z Asie, převážně Indie. Zákazníci platí společnosti za automatizované řešení CAPTCHA a ta je přeposílá online

pracovníkům, kteří tyto CAPTCHA řeší. Cena za 1000 vyřešených CAPTCHA se pohybuje kolem $1,5 a část zisku si samozřejmě ponechá zprostředkující společnost. Příkladem může být společnost Antigate.com, která uvádí, že průměrný čas pro vyřešení CAPTCHA je 15s od jejího zadání. V tomto případě se nejedná o prolomení CAPTCHA jako takové, jelikož je skutečně řešena lidmi, jde však o velmi účinné prolomění mechanizmu zabezpečení realizovaného pomocí CAPTCHA. Druhou možností je využít skript pro přeposlání CATPCHA na jiný webový server vlastněný útočníkem, často obsahující nelegální nebo pornografický obsah, kde nic netušící uživatel vyřeší CAPTCHA pro získání požadovaného obsahu a tím útočníkovi vyřeší CAPTCHA z původního serveru. Tento způsob je opět velmi účinný, jelikož je CAPTCHA řešena skutečnými lidmi, a také je zdarma. Jeho velkou nevýhodou však je omezení rychlosti řešení jednotlivých CAPTCHA, které je limitováno návštěvností utočníkova webu.

2.6.2 Chyby v implementaci

Dalším přístupem k překonání zabezpečení pomocí CAPTCHA je využít chyby v implementaci. Jednou z možností, jak udržovat uživatelský kontext u bezstavového protokolu HTTP, je použití Session ID [22], umožňující identifikaci sezení pomocí jedinečného identifikátoru. Častým problémem při implementaci CAPTCHA je však možnost obejít tohoto zabezpečení pomocí znovupoužití Session ID již známé CAPTCHA. Jinou možností je využití chyby, kdy část softwaru generujícího CAPTCHA je na klientské straně (validace je prováděna na serveru, ale text je renderovaný na straně uživatele). V tomto případě je možné klientskou část aplikace upravit, a tím zobrazit renderovaný text.

2.6.3 Vylepšené OCR

V předchozích dvou případech se nejednalo o prolomení CAPTCHA v pravém slova smyslu, ale o překonání ochrany, v rámci které se CAPTCHA používá. Chceme-li CAPTCHA prolomit, musíme být schopni správně rozpoznat znaky obsažené v obrázku. Jako hranice prolomení CAPTCHA se považuje 1%[2] úspěšnost správného rozpoznání všech znaků. Tato hodnota se zdá být nízká, uvažujeme-li ale rychlost, se kterou je možné jednotlivé CAPTCHA zkoušet, pak i 1% úspěšnost dává nezanedbatelné množství zneužití poskytované služby za hodinu.

První implementace CAPTCHA byly založeny pouze na mírně deformovaném textu v obrázku tak, aby nebyly čitelné běžnými programy pro rozpoznávání znaků (Optical Character Recognition, OCR). Díky tomu i metody k jejímu prolomení byly založeny na postupném vylepšování OCR softwaru.

Typický proces prolamování CAPTCHA se skládá z následnujících kroků:

1. **Předzpracování:** V této fázi dochází k odstranění pozadí pomocí různých algoritmů, obrázek je binarizován (převeden pouze na černou a bílou barvu) a uložen do matice binárních hodnot. Binarizace se provádí kvůli nižší paměťové náročnosti a vyšší rychlosti a snazší implementaci následných algoritmů. Binarizací obrázku sice ztrácíme informaci o intenzitě pixelu, v praxi to však nikdy nepůsobilo problémy.

2. **Segmentace:** V tomto kroku dochází k segmentaci jednotlivých znaků, tedy k jejich izolaci od zbytku obrázku. Často používanou technikou je CFS(Color Filling Segmentation), založené na semínkovém vyplňování. Výhodou této metody je její účinnost i na nakloněné znaky, pokud se nepřekrývají. U dutých CAPTCHA (pouze obrysy znaků) tato metoda funguje i pro překrývající se znaky.

3. **Post-segmentační úpravy:** V této části se jednotlivé segmenty zpracovávají samostatně, s cílem jejich přípravy pro snadnější rozpoznání. Mezi úpravy patří například normalizace jejich velikosti.

4. **Rozpoznání znaků:** Činnost v tomto kroku se dělí podle toho, zda se nacházíme ve fázi učení nebo testování. Ve fázi učení se klasifikátor učí, jak který znak po segmentaci vypadá. V testovací fázi následně klasifikátor predikuje, o jaký znak se pravděpodobně jedná. Klasifikátory, které se často používají jsou např. KNN(K Nearest Neighbors), tedy výběr na základě K nejbližších sousedů, SVM (Support Vector Machines), nebo klasifikátory založené na neuronových sítích, např. CNN(Convolutional Neural Network). První dvě zmíněné metody mají výhodu v rychlém učení, poslední, tedy CNN, pak v možnosti klasifikace i při rotaci a různé velikosti vzorku.

5. **Post-procesing:** V tomto kroku může docházet k dalším korekcím výstupu klasifikátoru. Jedná se především o opravu slov podle slovníku u CAPTCHA využívajích existujících slov.

Segmentace patří v procesu rozpoznání CAPTCHA k nejnáročnějším úkolům. Při správné segmentaci znaků nejsou rozdíly v úspěšnosti jednotlivých klasifikátorů příliš velké, je tedy výhodné použít klasifikátory s rychlým učením, díky čemuž jsme při vývoji schopni pozorovat efektivitu změn v krátkém čase, a ne až po desítkách minut či hodin, než se klasifikátor naučí požadovanou množinu znaků.

2.6.4 Algoritmy simulující činnost mozku

Poslední dobou se v souvislosti s prolamováním CAPTCHA objevují zprávy o použití algoritmů simulujících činnost mozku. Konkrétně se jedná o americkou společnost Vicarious. Ta se však nezabývá prolamováním CAPTCHA. Tato firma vyvíjí software, který nazývá rekurzivní kortikální síť[18], který je schopný myšlení a učení jako člověk, dokonce v takovém rozsahu, že je schopný představivosti. V nejbližších 5 letech však společnost nemá v plánu představit žádný finální produkt. Jejím dlouhodobým cílem je uplatnit její technologii v robotice či analýze obrazových medicínských dat. Rozpoznání CAPTCHA tedy představuje pouze demonstraci možností jejich softwaru. Tvrdí, že úspěšnost rozpoznání znaku v reCAPTCHA je u jejich softwaru 95%.

Společnost však bohužel neposkytuje žádné konkrétní informace o jejím produktu, pouze o svých pokrocích informuje prostřednictvím tiskových prohlášení, případně je prezentuje pomocí videí na svých webových stránkách.

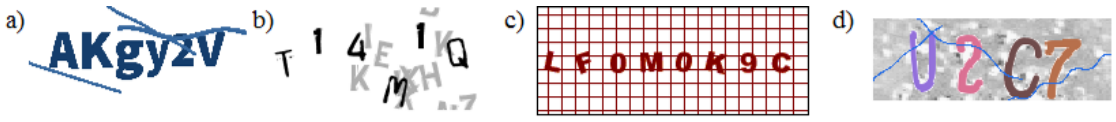
2.7 Proces generování

V této sekci si uvedeme typické kroky při generování textové CAPTCHA, její vývoj a doporučení pro generování silné CAPTCHA známá z předchozích prací. Jednotlivé kroky v procesu generování textové CAPTCHA typicky jsou:

- Definování množiny znaků
- Určení počtu znaků
- Generování řetězce znaků
- Vykreslení jednotlivých znaků

- Uplatnění technik pro zabránění rozpoznání znaků
- Uplatnění technik pro zabránění segmentace znaků

Prvních pět bodů je součástí procesu generování již od vzniku CAPTCHA, šestý bod se rozvíjí až v posledních letech, poté, co se zjistilo, že samotná deformace znaků či šum k zabránění rozpoznání znaků nestačí, a jako hlavní prostředek k zabránění prolomení CAPTCHA se začaly považovat právě techniky pro zabránění správné segmentace znaků.



Obrázek 2.12: Schémata prvních CAPTCHA; Způsoby snadného rozpoznání: a) odstranění čar na základě barevného filtru; b) odstranění matoucích znaků barevným filtrem; c) odstranění statické mřížky; d) segmentace znaků na základě barevného filtru

2.7.1 Vývoj CAPTCHA

První CAPTCHA byly velmi jednoduché a kladly si za cíl obstát pouze proti běžným programům pro strojové rozpoznání znaků (OCR). Jejich počet znaků byl často fixní, používaly stejný druh písma a pro oklamání OCR programů používaly často různé barvy znaků, barevné přechody, matoucí pozadí a později také přeškrtnutí znaků.

Obrázek 2.13: CAPTCHA společnosti Google

Použití různých barev se používalo velmi často, ve skutečnosti však žádný přínos pro složitost rozpoznání nemá, spíše naopak. Jak je vidět na příkladech z obrázku 2.12 , použití různých barev pro jednotlivé znaky usnadní jejich segmentaci použitím barevných filtrů. V případě použití jiné barvy pro znaky a jiné pro šum v pozadí opět pomocí barevného filtru usnadňuje odstranění šumu a tím velmi snadné provedení segmentace a rozpoznání znaků. Použití barevných přechodů se také míjelo účinkem, jelikož byly obrázky většinou převedeny do stupňů šedi či binarizovány. Jako matoucí pozadí se často používaly různé mřížky či čáry, které byly snadno detekovatelné a odstranitelné, čímž na obrázku zůstaly pouze snadno segmentovatelné znaky.

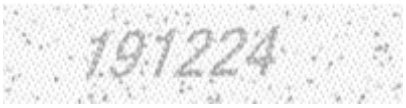
Vzhledem k tomu, že se začaly objevovat programy využívající upravené OCR, jež byly schopné si s těmito CAPTCHA poradit, začaly se CAPTCHA dále vyvíjet a vzniklo jich mnoho typů. Přestaly se používat různé barvy a obrázky se staly ve většině případů dvoubarevné. Začaly se také daleko více využívat techniky k zabránění segmentace a rozpoznání znaků. Znaky se začaly umisťovat různě do prostoru, ne pouze na jedné linii, často se používá také rotace znaků. K zabránění rozpoznání se nejčastěji používala lokální a globální

deformace znaků. Lokální deformací je myšleno deformování okrajů znaku (značně viditelné na obr. 2.1 na str. 6), globální pak deformace tvaru celého znaku (dobře viditelné např. na obr. 2.13). Ve značné míře se začal také používat šum pro zamezení segmentace znaků. Tím byly nejčastěji různě dlouhé, tlusté a zakřivené čáry, které měly sloužit k propojení jednotlivých znaků do jednoho objektu stejné barvy a k vyplnění mezer mezi znaky a tím zabránění segmentaci na základě projekce bodů do horizontální roviny (příklad na obr. 2.17). Příkladem mohou být CAPTCHA společnosti Google (obr. 2.13) ,Yahoo! (obr. 2.14) či serveru MSN (obr. 2.15) sloužící k zabezpečení registrace e-mailových schránek. Do té doby stačilo po odstranění pozadí provést Color Filling Segmentation (CFS) a díky tomu, že znaky se nedotýkaly, ani je nic nepropojovalo, byla segmentace znaků hotová.

Obrázek 2.14: CAPTCHA společnosti Yahoo!

Obrázek 2.15: CAPTCHA serveru MSN

Vznikaly také nové typy CAPTCHA, například takové, které se snažily znaky skrývat použitím podobných barev popředí a pozadí, či vytvořením CAPTCHA jako šum, kde byly znaky viditelné díky větší hustotě tmavších pixelů (obr. 2.16). Obě jsou však řešitelné na základě převodu obrázku do jiného barevného spektra či metodou založenou na hustotě pixelů.

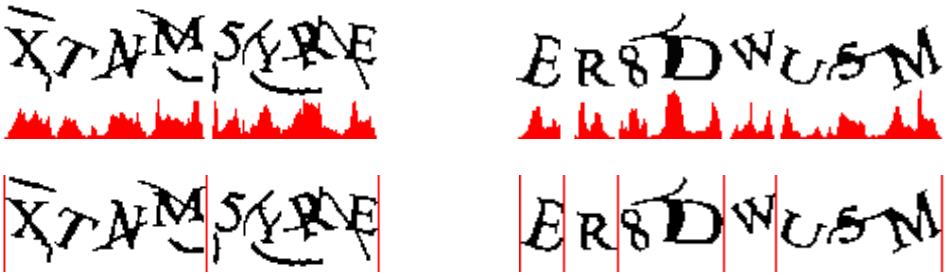


Obrázek 2.16: CAPTCHA využívající větší hustoty šumu pro vykreslení znaků

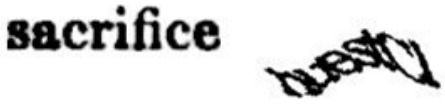
I přes tato opatření se stále vyvíjely programy, které byly CAPTCHA schopné prolomit, na což se reagovalo ještě větší deformací a větším množstvím šumu. To dospělo do stavu, kdy se jednotlivé CAPTCHA stávaly težko čitelné nejen pro programy, ale také pro lidi. Příkladem může být například staré schéma reCAPTCHA (viz obr. 2.18), které se stále používá při překročení určitého počtu zobrazení schématu nového.

V poslední době se tedy jednotlivé typy CAPTCHA zaměřují nejen na dostatečnou složitost automatického rozpoznání, ale také na vysokou úspěšnost jejich řešení lidmi. Novější CAPTCHA lze tedy poznat podle relativně snadného čtení znaků, přičemž zabránění prolomení je založeno na obtížné segmentaci znaků. Znaky se tedy často dotýkají, překrývají, sdílejí jejich části, či jsou umisťovány tak, aby se začátky a konce po sobě následujícch znaků překrývaly při zachování dobré čitelnosti znaků lidmi. Téměř samozřejmostí u těchto CAPTCHA je pak rozdílná velikost jednostlivých znaků, jejich různé natočení a poloha v obrázku. Často se také využívají různé typy písma.

Příkladem nového stylu CAPTCHA může být reCAPTCHA od společnosti Google (obr. 2.19), jež nedávno změnila schéma právě kvůli špatné úspěšnosti u lidí. Nyní je úkolem pře-



Obrázek 2.17: Vertikální projekce k segmentaci znaků



Obrázek 2.18: Staré špatně čitelné schéma reCAPTCHA

psat řetězec cifer, které jsou různě velké, pootočené a následující znaky mají překrývající se konce a začátky. Kvůli dobré úspěšnosti řešení lidmi jsou také vynechány cifry, které by mohly vést k nejednoznačnosti – 0-O, 1-7. Jak je u reCAPTCHA zvykem, tento řetězec je doplněn dalším obrázkem obsahující číslo k přepsání. Nyní jsou díky reCAPTCHA rozpoznávána čísla domů zachycená při snímání Google Street View.

Obrázek 2.19: Současné schéma reCAPTCHA

2.7.2 Doporučení pro tvorbu bezpečné CAPTCHA

Na základě předchozích prací lze uvést některá doporučení pro tvorbu bezpečné CAPTCHA.

Chceme-li vytvořit bezpečnou CAPTCHA, není nutné volit velkou množinu znaků, které se na ní mohou vyskytovat. Přesnost klasifikátoru při použití většího počtu možných znaků klesne jen o pár procent, současně s tím se však sníží také úspěšnost rozpoznání lidmi. Vhodné je také vyhnout se znakům, které se lidem při řešení CAPTCHA často pletou, např. i-j. Je tedy vhodné volit **spíše menší množinu nematoucích znaků**.

Daleko větší význam z hlediska bezpečnosti má počet znaků znaků v obrázku. Nemyslíme tím absolutní počet znaků, i když samozřejmě větší počet znaků snižuje procentuální úspěšnost rozpoznání, bohužel však také u lidí. Mnohem důležitější je proměnný počet znaků. Může-li se útočník spolehnout, že obrázek obsahuje vždy stejný počet znaků, je možné provést úspěšně segmentaci znaků i při použití jinak velmi účinných antisegmentačních technik.

Co se týká generování řetězce znaků, zde je doporučení velmi snadné – generovat náhodné řetězce znaků a nepoužívat existující slova. Při použití existujících slov je možné s celkem velkou přesností odhadnout původní slovo i přes špatné rozpoznání některých znaků.

Obrázek 2.20: CAPTCHA API od Seznam.cz

Pro ztížení segmentace a rozpoznání znaků je vhodné používat různé velikosti znaků a různé fonty. Zabráníme tím odhadům znaků na základě jejich velikosti, použití různých fontů také sníží přesnost klasifikátoru.

Techniky pro zabránění rozpoznání znaků by měly sloužit vždy jen k posílení bezpečnosti CAPTCHA, nelze se na ně spoléhat jako na způsob zabezpečení. Efektivita klasifikátorů je často závislá na velikosti a rotaci. Použití různých velikostí a rotace znaků je však vhodné kombinovat s anti-segmentačními technikami pro dosažení nepředvídatelnosti velikosti znaků při jejich segmentaci. Často se také používá deformace znaků. Jedná se o nejefektivnější způsob snížení úspěšnosti klasifikátorů, znatelně však snižuje úspěšnost rozpoznání také u lidí. Sama o sobě navíc není dostatečná k zabránění klasifikace znaků, je tedy vhodné se **deformaci znaků vyhnout** a nahradit ji vhodnými anti-segmentačními technikami.

Správná segmentace znaků je v současné době nejnáročnějším problémem v procesu prolamování CAPTCHA. Techniky pro zabránění segmentace znaků můžeme rozdělit do tří skupin – splynutí textu s pozadím, použití čar a překrývání znaků.

Splynutí s pozadím Pod tento pojem řadíme techniky, které se snaží zabránit segmentaci znaků pomocí jejich splývání s pozadím. Používají se tři hlavní způsoby, jak toho dosáhnout: použitím složitého pozadí, použitím pozadí s velmi podobnými barvami jako text a přidáním šumu do obrázku. Záměrem při použití **složitého pozadí** je promíchání čar a tvarů v pozadí s textem, a tím zabránění útočníkovi v nalezení a segmentaci znaků. Přestože dřívější práce [25] dokázaly, že tento typ ochrany není vetšinou bezpečný, stále se na něj některé CAPTCHA spoléhají. Většinou se používá náhodně generované pozadí, nebo části náhodných obrázků. V ojedinělých případech je použito pozadí statické, tedy na všech obrázcích stejné. Statické pozadí nemá z hlediska ochrany žádný význam, jelikož jej lze velmi snadno z obrázku odstranit. Statické pozadí je použito např. u CAPTCHA API společnosti Seznam.cz (obr. 2.20).

Podobným přístupem jako složité pozadí je přístup založený na **podobnosti barev** Ten vychází z použití takových barev, které jsou člověkem vnímané jako velmi rozdílné, ale ve skutečnosti jsou si v RGB spektru velmi podobné. Z hlediska prolomení však může být řešením převod obrázku do jiného spektra barev a následná binarizace s prahem určeným na základě hodnoty hue.

Poslední a nejúčinnější technikou je přidání **šumu** do obrázku. Šum však musí mít stejnou barvu jako text, jinak by jej bylo možné odstranit. V průběhu let však bylo navrženo mnoho technik pro odstranění šumu z CAPTCHA, proto ani tento přístup nelze považovat za účinný k zabránění segmentaci. Šum navíc snižuje úspěšnost rozpoznání znaků lidmi. V roce 2005 bylo dokonce dokázáno[3], že algoritmy založené na neuronových sítích mají při určování jednoho znaku menší chybovost než člověk. Na obrázku 2.21 vidíme úspěšnost rozpoznání znaků pro různě silné zašumění obrázků. Je vidět, že úspěšnost je velmi vysoká i u obrázků, kde už má člověk s rozpoznnáním znaku problém.

Typicky deformované znaky	Pravděpodobnost rozpoznání robotem
	~100%
	96+%
	100%
	98%
	~100%
	95+%

Obrázek 2.21: Úspěšnost rozpoznání znaků

Celkově se tedy metody založené na splynutí s pozadím nepovažují za bezpečné a použití pozadí je tedy doporučeno používat pouze ke kosmetickým účelům.

Použití čar Dalším přístupem k zabránění segmentace znaků je použití čar, které protínají více znaků. Mezi používanými CAPTCHA lze rozlišit dva typy čar používaných k zabránění segmentace. Prvním typem jsou čáry tenčí, než je tloušťka písma. Na základě této odlišné tloušťky lze však snadno oddělit čáry od znaků, čímž se jejich použití stává neúčiným. Vhodné je tedy použít druhý typ čar, tedy čáry se stejnou tloušťkou jako znaky, aby nebylo možné využít tloušťky k oddělení čar a znaků. Čáry také musí mít stejnou barvu jako znaky a je vhodné, aby délka čar byla proměnlivá, aby ani té nebylo možno využít k oddělení čar a znaků. Ke ztížení detekce čar je vhodné používat čáry, které neprotínají všechna písmena, a které mají sklon vyskytující se v částech znaků daného písma. V opačném případě by mohlo být k rozlišení čar a znaků použito právě sklonu čáry.

Překrývání znaků Překrývání znaků je považováno za nejúčinnější techniku proti segmentaci znaků. Její účinnost však může být omezena nevhodným návrhem jiných částí CAPTCHA. Je-li například překrývání znaků docíleno pouze odstraněním mezer mezi znaky, lze segmentaci znaků provést odhadem například na základě histogramu a průměrné šířky znaků. Je-li navíc počet znaků fixní, je segmentace ještě jednodušší. Pokud je však počet znaků proměnný, stejně tak jejich velikost a rotace, pak je segmentace znaků téměř nemožná a jedinou možností pro rozpoznání znaků je jejich rozpoznání bez předchozí segmentace.

Shrnutí Shrneme-li předchozí poznatky, pak pro vytvoření bezpečné CAPTCHA je vhodné použít řetězec znaků proměnlivé délky z ne příliš složité znakové sady bez matoucích znaků. Jednotlivé znaky pak vykreslit různými fonty a v různé velikosti. Znaky je dobré umístit tak, aby se překrývaly, a přeškrtnout je čárou stejné barvy a tloušťky. Celý obrázek je také vhodné zvlnit, pro ztížení detekce čar. Účinné je také vytvářet alternativní schémata, kterými je možné současné schéma v případě prolomení CAPTCHA okamžitě nahradit nebo je také čas od času obměnit, a tím zmařit dočasnou práci útočníků na aktuálním schématu.

Kapitola 3

Analýza problému

Jak již bylo zmíněno v předchozí kapitole, nejnovější CAPTCHA mají svoji bezpečnost založenou hlavně na antisegmentačních technikách. Díky tomu je možné obrázek CAPTCHA předzpracovat za pomoci několika jednoduchých funkcí a přejít k hlavnímu problému, tedy segmentaci znaků.

3.1 Určení problému

Jako hlavní cíl této práce bylo zvoleno prolomení reCAPTCHA, tedy implementace CAPTCHA vlastněnou od roku 2009 společností Google. Důvod zvolení právě této CAPTCHA je zřejmý – jde pravděpodobně o nejrozšířenější a nejpoužívanější CAPTCHA na internetu, je zaštítěna velkou společností Google, je poskytována zdarma a uživatel při jejím řešení pouze nemrhá svým časem, ale přispívá k digitalizaci knih či zdokonalování map společnosti Google.

Jejím prolomením tedy narušíme zabezpečení tisíců služeb na internetu a díky jejímu použití také při zakládání e-mailových účtů GMail.com společnosti Google získáme také možnost registrace těchto ceněných a spamovými filtry méně filtrovaných e-mailových účtů.

Pro zmenšení dopadu prolomení je počet zobrazení nového schématu reCAPTCHA omezen a následně je zobrazováno schéma starší, špatně čitelné i lidmi. Omezený počet zobrazení daného schématu však může případný útočník snadno obejít použitím tzv. botnetu, tedy sítě napadených a ovládaných počítačů [20], pomocí kterých je možné během několika sekund provést ohromné množství úspěšných žádostí o danou službu. Hlídání počtu použití daného schématu je vztaženo k určitému identifikátoru počítače, při jeho podvrhnutí je tedy teoreticky možné provádět z jednoho zařízení neomezený počet žádostí o danou chráněnou službu.

Možnost použití vytvořené techniky demonstrujeme také na CAPTCHA API od společnosti Seznam.cz a vybraných schématech z captcha.com.

3.2 Analýza reCAPTCHA

Kvůli složitosti a nízké úspěšnosti u lidí bylo schéma reCAPTCHA omezeno z přepisování slov pouze na číslice. Kvůli možné nejednoznačnosti u znaků O-0 a 1-7 jsou tyto znaky vynechány. Obrázek reCAPTCHA (obr. 3.1) se skládá ze dvou částí. V jedné části je řetězec z číslic 2,3,4,5,6,8,9 dlouhý 6-8 znaků, v druhé části je obrázek s číslem domu pořízený z Google Street View. Jelikož číslo domu není známé a cílem je jej po vyřešení CAPTCHA

získat, je možné na tomto místě napsat cokoliv, aniž by to ovlivnilo úspěšnost vyřešení CAPTCHA. Tento obrázek tedy můžeme bez problému ignorovat. Pořadí zmíněných částí se může měnit, také jejich předěl není vždy přesně uprostřed, nepotřebný obrázek lze však snadno detekovat a z obrázku odstranit.

Obrázek 3.1: Současné schéma reCAPTCHA

Kontrolní část obrázku neobsahuje žádný šum ani čáry protínající jednotlivé znaky. Jako antisegmentační technika je použito různé vertikální umístění znaků, různá velikost znaků, rotace znaků, ale hlavně jejich dotyk a překryv obdélníků vymezujících jejich polohu.

3.2.1 Návrh segmentační techniky

Díky překryvu a dotyku znaků není použití CFS a oddělení znaků na základě histogramu po projekci do horizontální roviny samostatně použitelné. Jako možný způsob segmentace znaků jsme zvolili kombinaci několika přístupů, které se používaly na některé starší CAPTCHA a metody používané při rozpoznávání čínského textu.

Jako základní metody pro segmentaci znaků použijeme nelineární řez na základě nejkratší cesty [7] a již zmíněné CFS (cite) a použití histogramu po projekci do horizontální roviny (cite).

Nelineární řez vychází ze segmentace lehce se dotýkajících znaků v obrázku ve stupních šedi. U toho se předpokládá, že intenzita bílé barvy na okraji znaku, a tedy i mezi jednotlivými znaky, bude nižší, než uprostřed znaku. Díky tomu je možné hledáním nejkratší cesty ve vhodně zvolené části obrázku provést oddělení obou znaků, přičemž za nejkratší cestu se považuje taková cesta od horního pixelu k dolnímu pixelu ve vybrané části obrázku, která má nejnižší součet intenzit pixelů na dané cestě. Jako vhodně zvolená část obrázku se považuje pravděpodobný předěl mezi znaky.

Color Filling Segmentation (CFS) slouží k segmentaci obrázku na základě spojitých objektů. Využívá se při něm tzv. semínkového vyplňování, kdy se za spojitý objekt považuje taková část obrázku, která je celé vyplněna při semínkovém vyplňování započatém v jakémkoli jejím bodě. U jednodušších CAPTCHA stačilo CFS k samotné segmentaci znaků, díky jejich dotyku jej však využijeme pro pomocné rozdělení obrázku na samostatné objekty.

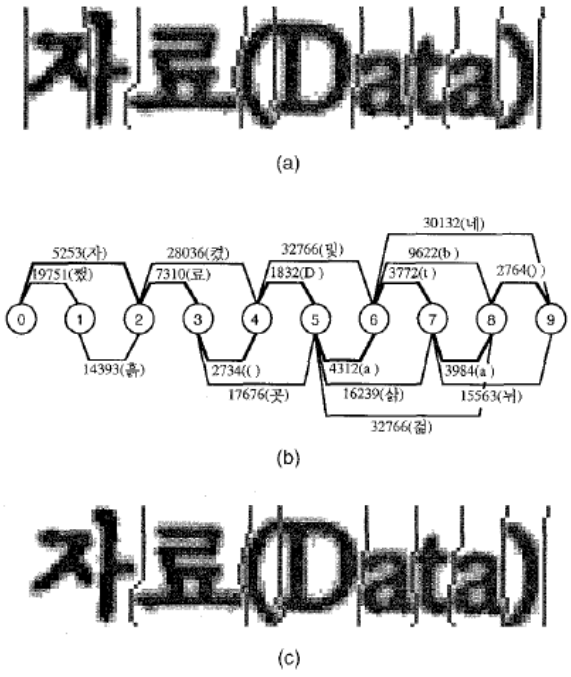
Histogramu po projekci do horizontální roviny se často využívalo pro segmentaci oddělených, či lehce se dotýkajících znaků, bez překryvu obdélníků vymezujících jejich umístění v obrázku. Tato metoda je založena na projekci bodů do horizontální roviny, čím dojde k "setřepání"pixelů směrem dolů a součtu intenzit jednotlivých bodů v každém sloupci obrázku. Tím získáme zmiňovaný histogram odpovídající celkové intenzitě pixelů pro každý sloupec. Jsou-li znaky v obrázku od sebe odděleny mezerou, je tato mezera v histogramu představována nulovými hodnotami. U lehce se dotýkajících znaků lze jejich předěl nalézt pomocí výrazného minima v příslušném sloupci v histogramu.

Kombinací těchto metod získáme nejprve samostatné spojité objekty pomocí CFS. Nad těmito objekty následně provedeme projekci do horizontální roviny. Ve vzniklých histogramech najdeme všechna minima, která budou sloužit jako pravděpodobné předěly mezi znaky. Pro vybrané minimum pak vždy najdeme střed mezi jeho x-ovou souřadnicí a x-

ovou souřadnicí minima předchozího a následujícího. Tím získáme oblast pravděpodobného předělu mezi znaky pro hledání nejkratší cesty nelineárním řezem.

Díky tomu, že předěl znaků již nehledáme podle nulové hodnoty v histogramu, jak tomu bylo u jednodušších CAPTCHA, ale využíváme minim v histogramu, dostáváme spoustu řezů i v jiných místech, než ve skutečných předělech znaků. Příkladem může být znak 0, kdy se minimum v histogramu objevuje i ve středu tohoto znaku. Ze získaných řezů tedy musíme vybrat ty nejpravděpodobnější.

K tomuto účelu využijeme techniku rozpoznáním řízené segmetnace, používanou například při segmentaci textu obsahujícího čínské znaky, které se mohou skládat z několika samostatných objektů. Tato technika je založena na využití rozpoznání znaků již při jejich segmentaci. Při rozpoznávání takového textu vznikají také kandidátní řezy, z nichž je potřeba vybrat ty nejvhodnější. K tomuto účelu se určí maximální šířka znaku, kterou může nabývat, a jednotlivé části obrázku vzniklé řezy se spojují do všech kombinací po sobě následujících částí, jenž tuto velikost nepřekračují. Všechny takto vzniklé kombinace se pokusíme rozpoznat klasifikátorem a uložíme si pravděpodobný výsledek i se vzdáleností od jeho vzoru. Vzhledem k tomu, že požadujeme využití všech částí obrázku právě jednou, budeme hledat takovou kombinaci použitých kombinací, jejichž celkový součet vzdáleností je nejmenší. Tento problém tedy převedeme na problém hledání nejkratší cesty v grafu (viz obr. 3.2), kde výchozím uzlem bude uzel reprezentující nepoužití žádné části obrázku, a koncovým uzlem bude uzel reprezentující použití všech částí obrázku právě jednou. Jednotlivé hrany grafu budou ohodnoceny podle vzdálenosti získané z klasifikátoru pro danou část obrázku. Hledání nejkratší cesty navíc omezíme tak, aby cesta vedla přes požadovaný počet uzlů – tedy rozpoznaných znaků. Výsledkem nalezení této cesty je již také samotné rozpoznání znaků v takto segmentovaném obrázku.



Obrázek 3.2: Optimální segmentace znaků: a) Kandidátní nelineární řezy, b) Segmentační graf, c) Optimální nelineární segmentace znaků, zdroj [7]

Kapitola 4

Návrh aplikace

V této kapitole se seznámíme s návrhem aplikace, která bude v rámci této práce vytvořena.

Bude se jednat o multiplatformní desktopovou aplikaci implementovanou v prostředí Qt/C++, jejímž cílem bude rozpoznávání různých druhů CAPTCHA a poskytování prostředí pro testování a ladění použitých algoritmů. Aplikace bude umožňovat zpracování konkrétní CAPTCHA jejím načtením ze souboru nebo jejím přetažením do okna aplikace, případně dávkové testování všech obrázků ze složky.

Vzhledem k tomu, že reCAPTCHA není jediným typem CAPTCHA, který má vytvořená aplikace řešit, byl její návrh přizpůsoben univerzálnímu použití pro různé typy CAPTCHA. Aplikace bude obsahovat různé metody pro zpracování obrázku rozdělené do typických fází procesu rozpoznávání CAPTCHA, tedy předzpracování, segmentace, postsegmentační úpravy a rozpoznání. V každé této skupině bude možné vybírat z dostupných metod ty, které budou pro vybraný typ CAPTCHA vhodné. Metody bude možné zařarovat do procesu i opakovaně a měnit pořadí jejich provádění. Při výběru použité metody se také zobrazí její dostupné parametry pro jejich nastavení. V nastavení klasifikátoru bude také možné zadat znaky, které může CAPTCHA obsahovat, adresář, ve kterém budou hledány vzory těchto znaků, a také minimální a maximální délku hledaného řetězce znaků.

Pro snadnou vizuální kontrolu jednotlivých kroků budou zobrazeny výstupy z jednotlivých metod při zpracování obrázku. Pro hromadné testování více obrázků bude možné vybrat složku, z niž budou otestovány všechny obrázky a výsledek rozpoznání porovnán s názvem souboru. Následně bude zobrazena také celková úspěšnost, čas zpracování obrázků a výpis testovaných obrázků s výsledkem rozpoznání.

Jednotlivé metody včetně jejich nastavení si bude aplikace pamatovat i po jejím ukončení. Veškerá nastavení procesu rozpoznání se budou ukládat podle zadaného názvu zkoumané CAPTCHA, bude tedy možné přidávat nové způsoby zpracování a rozpoznání pro jiné typy CAPTCHA a mezi těmito nastaveními se přepínat.

Pro snadnější přípravu vzorových obrázků pro rozpoznání bude možné uložit výstupní obrázek z fáze předzpracování. Díky tomu bude snazší provést ruční segmentaci jednotlivých znaků a jejich uložení jako vzory pro klasifikátor.

Kapitola 5

Popis implementace

V této kapitole se seznámíme s detaily týkajícími se implementace aplikace, konkrétně popisem grafického uživatelského rozhranní aplikace a s implementovanými metodami pro zpracování a rozpoznání CAPTCHA.

Vytvořená aplikace je implementována v prostředí Qt/C++, jde tedy o multiplatformní aplikaci. Pro správný překlad a běh aplikace je nutné mít nainstalovány knihovny Qt verze 4.8 a OpenCV 2.4.8.

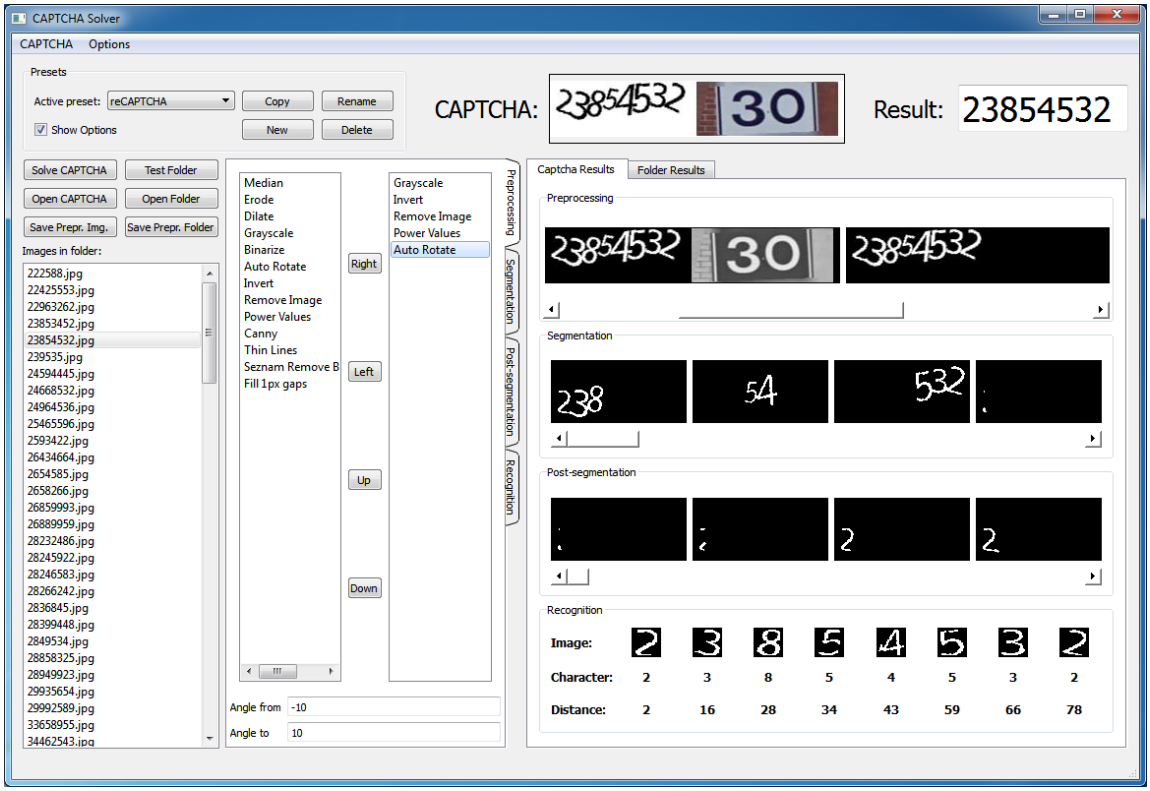
5.1 Grafické uživatelské rozhranní

Hlavní okno aplikace (obr. 5.1) je rozděleno do několika částí. V levé horní části se nachází nastavení presetů, tedy uložených nastavení pro jednotlivé typy CAPTCHA. Ty je možné kopírovat, přidávat nové, přejmenovat a mazat. Ve střední horní části je zobrazena vybraná CAPTCHA, vpravo pak poslední výsledek rozpoznání.

Spodní část aplikace je rozdělena do tří sloupců. V levém z nich je možné provádět operace s obrázky a složkami, konkrétně tedy řešit vybraný obrázek/složku, otevřít obrázek/složku a uložit předzpracovaný obrázek/složku. Při výběru operace nad složkou je tato operace provedena nad všemi obrázky v aktuálně vybrané složce. Pod tlačítky výběru akce je zobrazen seznam všech obrázků z aktuální složky. Výberem obrázku dojde k jeho zobrazení v horní části okna, dvojklikem na jeho název dojde k jeho řešení.

V prostředním sloupci se nachází veškeré nastavení týkající se zpracování CAPTCHA. Je možné vybrat ze 4 záložek pro nastavení požadované fáze. Pro fáze předzpracování, segmentaci a post-segmentaci jsou zobrazeny dva seznamy. V levém jsou zobrazeny dostupné metody pro vybranou fázi, v pravém jsou zobrazeny vybrané metody, které se při zpracování budou provádět v pořadí shora dolů. Výběrem metody v pravém sloupci dojde také k zobrazení dostupných nastavení (viz dolní část obr. 5.1) pro jejich okamžitou změnu.

V záložce nastavení rozpoznání (levá část obr. 5.2) je možné povolit či zakázat provádění rozpoznávání znaků. Dále je možné vybrat složku se vzorovými obrázky, seznam znaků, které se mohou v obrázku vyskytnout a příponu vzorových obrázků. Ty musí být pojmenované ve formátu [znak]_[pořadí][.přípona], kde [znak] je jeden ze seznamu znaků vyskytujících se v daném typu CAPTCHA, [pořadí] je pořadové číslo vzorového obrázku, které je větší rovno 1, a [přípona] značí příponu vzorového obrázku. Jména obrázků tedy mohou být například A_1.jpg, A_2.jpg, atd. Při přerušení posloupnosti čísel [pořadí] se přechází na hledání dalšího znaku. Dále je možné nastavit počet použitých vzorových obrázků pro učení klasifikátoru, je-li zvolena hodnota 0, použijí se všechny až do přerušení posloup-



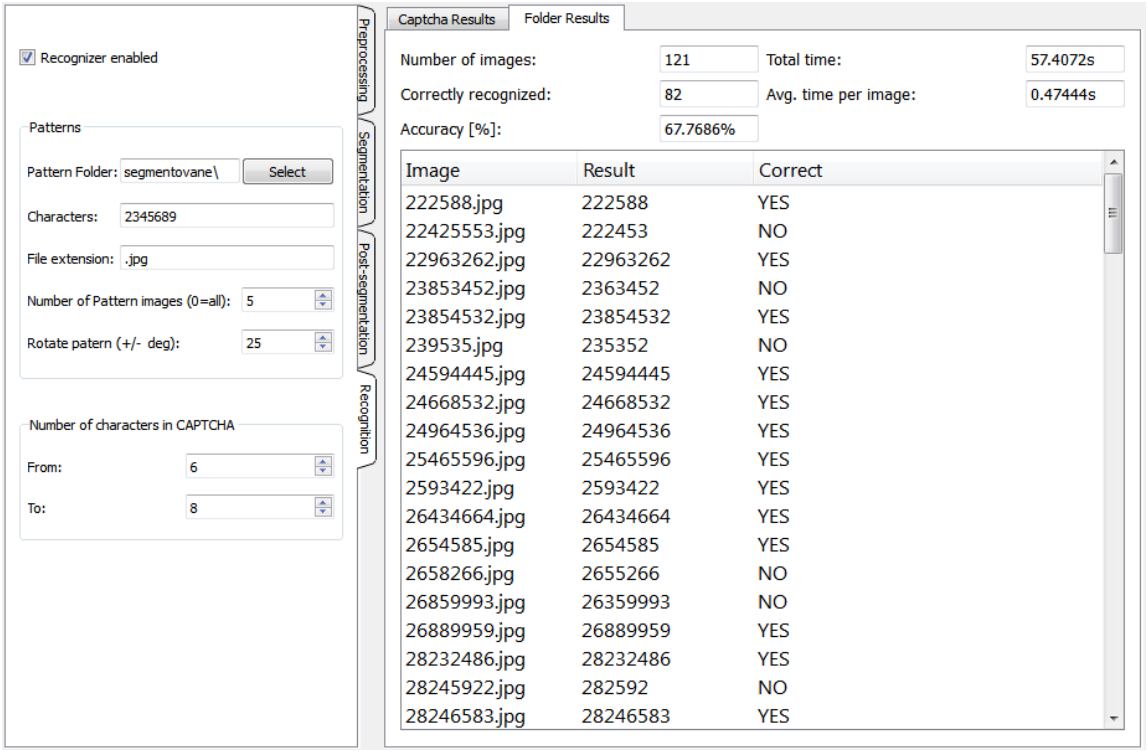
Obrázek 5.1: GUI aplikace

nosti [pořadí], je-li zvolena hodnota větší než 0, použije se prvních n obrázků. Dále je možné zvolit rozmezí úhlů, o které bude vzorový obrázek postupně rotován a ukládán do klasifikátoru. Posledním nastavením týkajícím se rozpoznávání je minimální a maximální počet znaků vyskytujících se v daném typu CAPTCHA.

Tyto záložky s nastavením je možné také skrýt a opětovně zobrazit, čímž získáme více prostoru pro zobrazení výsledků jednotlivých metod. Provedeme tak dvojklikem na záložku, či použitím zaškrtávacího políčka "Show Options"v levé horní částí aplikace.

Třetím sloupcem, a také poslední částí aplikace, je prostor k zobrazení výsledků rozpoznání. Pomocí záložek je možné zvolit mezi zobrazením výsledků rozpoznání jedné CAPTCHA a zobrazením výsledků hromadného testování celé složky. Při zobrazení výsledků zpracování jednoho obrázku jsou zobrazeny výstupy jednotlivých použitých metod při zpracování obrázku, ty jsou opět seskupeny podle jednotlivých fází. Ve výsledcích fáze rozpoznání je kromě použitých a normalizovanch segmentů zobrazen také výsledek rozpoznání a vzdálenost od nejbližšího vzoru.

Přepnutím na záložku hromadných výsledků získáme zobrazení výsledků posledního testování všech obrázků z vybrané složky (viz obr 5.2). Zde je zobrazen celkový počet úspěšně otevřených obrázků, počet správně rozpoznaných obrázků a procentuální úspěšnost rozpoznání. Dále je zobrazen celkový čas zpracování všech obrázků a průměrný čas zpracování jednoho obrázku. Pod těmito informacemi je pak zobrazen seznam všech úspěšně otevřených obrázků spolu s výsledkem jejich rozpoznání a informací o tom, zda bylo rozpoznání správné. Dvojklikem na záznam o obrázku dojde k výběru a řešení vybraného obrázku, což umožní zobrazení výstupů jednotlivých fází rozpoznání a tím kontrolu, proč nedošlo k správnému rozpoznání.



Obrázek 5.2: Detail nastavení rozpoznání a výsledků testování složky

Délka provedení hromadného testu může v závislosti na nastavení a počtu testovaných obrázků dosáhnout několika desítek sekund. Jelikož není grafické rozhraní aplikace implementováno v samostatném vláknu, aplikace po tuto dobu nereaguje a může se tvářit, že zamrzla. Po dokončení testu posledního obrázku se však zobrazí výsledky a vše funguje, jak má. O probíhajícím výpočtu informuje poznámka ve stavovém řádku aplikace.

5.2 Implementované metody

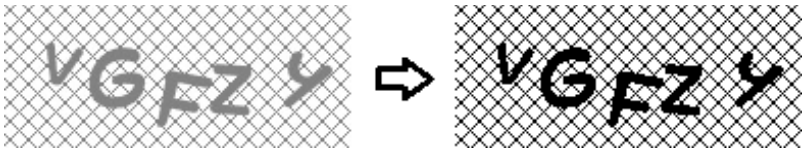
Jak bylo v návrhu aplikace zmíněno, vytvořená aplikace neslouží pouze pro rozpoznání reCAPTCHA, implementované metody zpracování obrázku v jednotlivých fázích tedy neobsahují pouze metody nutné potřebné pro rozpoznání tohoto typu CAPTCHA, ale také další, které by se mohly hodit pro zpracování i jiných typů.

5.2.1 Předzpracování

Cílem fáze předzpracování obrázku je dostat jej do stavu, kdy na obrázku zůstanou pokud možno pouze samotné znaky určené k segmentaci. Z požadavku, aby aplikace byla schopná pracovat s různými typy CAPTCHA, logicky vyplývá, že výčet metod v tého sekci bude největší.

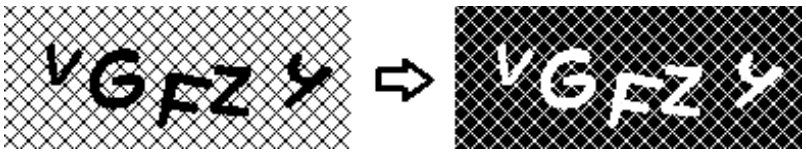
Jelikož většina dalších metod pracuje s obrázky ve stupních šedi, je vhodné obrázek nejprve převést do tohoto formátu metodou **Grayscale**. Převod obrázku do stupňů šedi není prováděn automaticky, kvůli možné pozdější implementaci metod, které informací o barvách využívají.

Další metodou je **Binarizace** obrázku (obr. 5.3). Při té se zjistí minimální a maximální intenzita obrázku a její střed se použije pro binarizaci obrázku metodou prahování. Pixelům s intenzitou nižší než tento práh se přidělí intenzita 0, tedy černá, pixelům s vyšší intenzitou je nastavena hodnota 255, tedy bílá.



Obrázek 5.3: Binarizace obrázku

Jelikož algoritmy pro segmentaci znaků fungují na základě předpokladu, že popředí, tedy znak, má vyšší intenzitu než pozadí, je v některých případech nutné provést inverzi barev obrázku metodou **Invert** (obr. 5.4).



Obrázek 5.4: Inverze barev obrázku

Pro jednoduché odstranění šumu je možné využít metody **Medián**, která aplikuje mediánový filtr se zadanou velikostí matice. Na obrázku 5.5 vidíme ukázku použití mediánového filtru s velikostí zkoumané matice 5, díky čemuž dojde k odstranění tenkých čar v pozadí.

Obrázek 5.5: Mediánový filtr s velikostí matice 5

Další možností, jak odstranit šum či tenké čáry, je využít metod **Erode** a **Dilate**. Metoda Erode způsobí odebrání 1px od okrajů objektů, způsobí tedy zmizení tenkých čar a ztenčení širších objektů (viz obr. 5.6). Metoda Dilate funguje přesně opačně oproti Erode. Dojde tedy k rozšíření objektů do všech stran (viz obr. 5.7).

Obrázek 5.6: Odebrání okrajových pixelů objektů metodou Erode

Je-li pozadí obrázku statické, jako v případě CAPTCHA API od Seznam.cz v přechozích příkladech, je možné pozadí z obrázku odstranit také jeho odečtením. K tomu můžeme použít metodu **Remove Background**, které jako parametr nastavíme cestu k obrázku

Obrázek 5.7: Rozšíření objektů do všech stran metodou Dilate

s pozadím. Metoda neprovádí pouhé odečtení hodnot intenzit pixelů, ale také kontroluje, zda by po odečtení pozadí nevznikla díra uprostřed znaku, v takovém případě se pozadí u daného pixelu neodečte a tím je zachována celistvost znaků v popředí (viz obr. 5.8).

Obrázek 5.8: Odstranění statického pozadí jeho odečtením se zachováním celistvosti popředí

V případě potřeby ztenčení objektů až na šířku 1px je možné využít metody **Thin Lines**, čímž získáme v podstatě kostru jednotlivých objektů v obrázku (viz obr. 5.9).

Obrázek 5.9: Nalezení kostry objektů pomocí ztenčení čar metodou Thin Lines

Pro nalezení obrysů objektů je možné využít metodu **Canny** imeplementující Canny detektor [21] pro nalezení hran. Parametrem této metody je spodní práh, pomocí kterého lze ovlivnit počet nalezených hran.

Obrázek 5.10: Canny detektor pro nalezení hran

Některé typy CAPTCHA obsahují znaky rozřezané na malé kostičky či obdélníčky, které člověku při pohledu z větší vzdálenosti splývají a vnímá tak celé znaky. Jsou-li mezery mezi obdélníčky šířky 1px, je možné použít metodu **Fill 1px gaps** pro jejich vyplnění (obr. 5.11). Tato metoda zkoumá všechny pixely s barvou pozadí, zda mají oba sousední pixely v horizontálním či vertikálním směru barvu popředí. Pokud ano, jedná se o 1px mezeru a ta je následně vyplněna barvou popředí. V případě, že jsou mezery mezi obdélníčky větší, je možné použít metodu **Dilate**.

Při práci s obrázky ve stupních šedi je také možné využít metody **Power Values**. Tato funkce převede hodnoty pixelů z intervalu 0-255 do intervalu 0-1, poté je umocní na zadaný

Obrázek 5.11: Fill 1px gaps pro vyplnění 1px mezer

exponent a převede zpět do intervalu 0-255. Této metody lze využít k zvýraznění předělů mezi znaky (obr. 5.12), které mají o něco nižší intenzitu, než znaky samotné. Umocněním těchto hodnot na exponent větší než 1 se tmavší pixely ztmaví ještě více, přičemž bílé pixely zůstanou stále plně bílé. Zadáním exponentu v intervalu 0-1 dojde naopak k zesvětlení tmavých pixelů a tím zesvětlení obrázku. Toho je možné využít při načítání obrázků ve formátu GIF, který není knihovnou OpenCV podporován, a pro jeho otevření je použita knihovna GifLib, která z neobjevených příčin načítá obrázek velmi tmavý.

Obrázek 5.12: Metoda Power Values pro zvýraznění přechodů mezi znaky

Jak bylo zmíněno v sekci 3.2.1, při segmentaci využíváme minim v histogramu po projekci do horizontální roviny k určení pravděpodobných předělů mezi znaky. Jelikož jsou u některých obrázků reCAPTCHA všechny znaky mírně rotovány určitým směrem, je vhodné i zmíněnou projekci provádět pod určitým úhlem. K tomu můžeme využít metody **Auto Rotate** (obr. 5.13), která provádí postupnou rotaci obrázku v rozmezí úhlů zadaných jako parametr. Jelikož je cílem v histogramu najít minimální počet minim, která budou mít co největší rozdíl od maxim v histogramu, ohodnotíme jednotlivé rotované obrázky průměrnou hodnotou rozdílů sousedních minim a maxim. Jako výsledek je tedy vybrán úhel s největším průměrem. Na konci je výsledný obrázek oříznut pro odstranění prázdného prostoru, který bylý přidán kvůli rotaci.

Obrázek 5.13: Metoda Auto Rotate pro nalezení optimálního pootočení obrázku kvůli úspěšnější segmentaci znaků

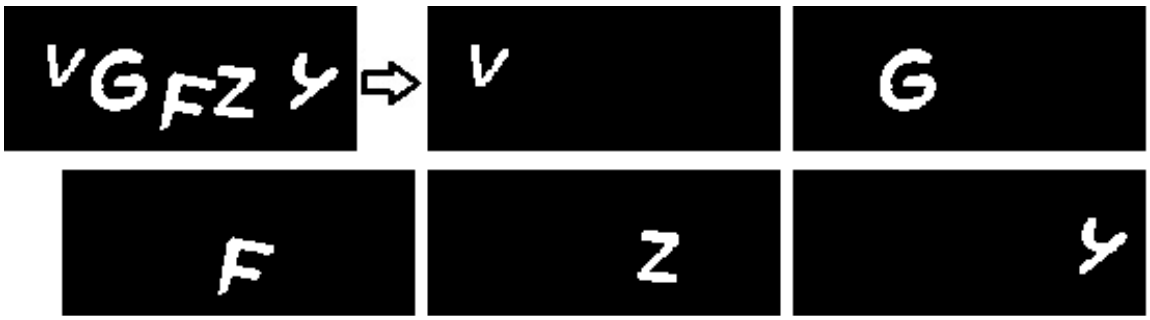
Speciálně pro reCAPTCHA byla vytvořena vytvořena metoda **Remove Image** (obr. 5.14) sloužící k detekci a odstranění doplňujícího obrázku, který nemá pro správnost řešení žádný vliv. Tato metoda převede všechny pixely s intenzitou větší než 0 na pixely s intenzitou 255. Následně ve střední části obrázku nalezne hranu doplňujícího obrázku nalezením největšího skoku v histogramu po projekci do horizontální roviny, a světlejší část obrázku od této hrany smaže.

5.2.2 Segmentace

Pro fázi segmentace jsou implementovány 2 metody.

Obrázek 5.14: Metoda Remove Image pro detekci a odstranění doplňujícího obrázku z reCAPTCHA

První z nich je **Color Filling Segmentation**, sloužící k rozdělení obrázku na samostatné spojité objekty (obr, 5.15). Tato metoda hledá v obrázku pixel s barvou popředí. Po jeho nalezení provede semínkové vyplňování započaté z tohoto bodu a při jeho provádění odstraňuje procházené pixely z původního obrázku a zapisuje je do nového. Každý spojitý objekt tedy zapíše do samostatného obrázku. Takto se pokračuje, dokud není původní obrázek prázdný.



Obrázek 5.15: Metoda Color Filling Segmentation

Druhou metodou je **nelineární řez podle histogramu**. V této metodě se provede projekce obrázku do horizontální roviny a ve vzniklém histogramu se naleznou minima. Tato minima poslouží k určení pravděpodobných předělů mezi znaky, ve kterých se bude provádět nelineární řez. Pro každé minumum je tedy definován prostor pro následný řez. Tento prostor začíná v polovině mezi x-ovou souřadnicí předcházejícího minima a x-ovou souřadnicí aktuálního minima. Konec prostoru je určen stejným způsobem, ale vzhledem k následujícímu minimu. Samotný řez poté probíhá jako hledání nejkratší cesty mezi pixely z horního řádku definovaného prostoru do spodního řádku prostoru [7], přičemž z bodu [x,y] jsou povoleny přechody pouze do bodů [x-1,y+1], [x,y+1] a [x+1,y+1], a vzdálenosti mezi body jsou určeny intenzitou jednotlivých pixelů. U této metody je také možné nastavit minimální šířku a výšku souvislého objektu, které určují, že se budou řezy provádět pouze na objektech, které alespoň jednu z hodnot překračují.

5.2.3 Post-segmentace

Fáze post-segmentace slouží k přípravě segmentovaných znaků pro rozpoznání. Dochází v ní k normalizaci velikosti rozpoznávaného obrázku. Jelikož je tento proces nutný před každým rozpoznáním, není zde možnost jej zařazovat nebo odebírat z procesu zpracování CAPTCHA a místo toho je automaticky prováděn těsně před fází rozpoznání.

Jedinou metodou, kterou je v této fázi možné použít, je metoda **Join Images**. Ta slouží ke spojování po sobě následujících řezů až do zadané maximální šířky. Tuto metodu je vhodné použít při použití metody nelineárních řezů, jelikož se minima v histogramu nenacházejí pouze na předělu znaků, ale také uvnitř znaků, čímž dochází k rozřezání samotných

znaků na několik částí. Tato metoda tedy slouží k pospojování těchto částí opět do podoby celého znaku. Metoda má dva parametry, jedním lze nastavit maximální šířku vzniklého obrázku, druhým parametrem lze nastavit, zda se mají řezy spojovat lineárně, či nikoliv. Lineární spojování znamená, že z řezu číslo n vzniknou obrázky jeho spojením s řezem $n+1$, poté $n+1$ a $n+2$, a tak dále, dokud jsou výsledné obrázky menší, než zadaná maximální velikost. Nelineární spojování znamená, že vzniknou obrázky spojením všech kombinací řezů, které budou mít velikost menší, než zadané maximum. Nelineární spojování má tedy smysl např. po segmentaci pouze metodou CFS, kdy nemáme zaručeno lineární pořadí jednotlivých objektů, jako je tomu při segmentaci použitím nelineárních řezů.

5.2.4 Rozpoznání

Jelikož je aplikace určena spíše pro nalezení způsobu prolomení různých typů CAPTCHA, než nasazení pro prolamování spousty obrázků jednoho druhu CAPTCHA, předpokládají se časté změny procesu rozpoznání, včetně změn parametrů použitých metod a nastavení klasifikátoru. Pro uživatele je tak důležité, aby dostal výsledky po provedených změnách co nejdříve a nemusel čekat v řádech minut na dokonalé naučení neuronové sítě či jiného klasifikátoru.

Z tohoto důvodu byl jako klasifikátor pro rozpoznání znaků zvolen K-Nearest Neighbours (KNN) [12], který vyniká rychlým učením i klasifikací. V našem případě budeme hledat pouze nejbližšího souseda, tedy K bude rovno 1. Podobně jednoduché řešení jsme zvolili také pro volbu vektoru rysů obrázku. Ten jednoduše získáme linearizací jednotlivých řádků obrázku do výsledného vektoru. Výsledkem procesu klasifikace je tedy třída a vzdálenost od nejbližšího vzoru.

V předchozí fázi procesu rozpoznání probíhá spojování jednotlivých řezů. Do fáze rozpoznání tedy vstupuje množina obrázků vzniklých v předchozí fázi, spolu s informací o tom, které řezy obrázek obsahuje. Tyto obrázky následně vstupují do klasifikátoru, čímž jsou doplněny o informaci o nejbližší třídě a vzdálenosti k nejbližšímu vzoru. Z těchto dat následně vytváříme graf, v němž budeme hledat nejkratší cestu pro nalezení nejlepšího výsledku.

Jednotlivé uzly grafu jsou identifikovány množinami řezů, které obsahují. Počáteční uzel tedy bude identifikován prázdnou množinou použitých řezů, cílový uzel bude identifikován množinou obsahující všechny řezy. Vzhledem k tomu, že cílový uzel musí obsahovat všechny řezy, můžeme graf vytvořit jako orientovaný graf, kde jednotlivé hrany z daného uzlu povedou vždy do uzlu s nejnižším nepoužitým číslem řezu, čímž omezíme větvení stromu. Ohodnocení jednotlivých hran je dáno vzdálenostmi od nejbližšího vzoru. Jako algoritmus pro hledání nejkratší cesty v grafu byl zvolen Dijkstrův algoritmus [4].

Vzhledem k omezení na minimální a maximální počet znaků v obrázku je nutné algoritmus upravit. První modifikace algoritmu byla založena na tom, že uzly neobsahovaly pouze hodnotu nejkratší cesty a ukazatel na předchůdce, ale obsahovaly všechny délky všech cest, kterými je možné se do uzlu dostat, spolu s ukazateli na předchůdce. To umožňovalo ignorování nalezené cesty v případě, že procházela přes malý nebo velký počet uzlů, čímž došlo k hledání další nejkratší cesty a tím pozdějšímu nalezení cesty s požadovanou délkou. Tato modifikace však způsobila vysokou paměťovou i časovou náročnosti algoritmu, kdy k vyřešení jedné CAPTCHA trvalo průměrně 7s.

Uzly grafu byly tedy následně modifikovány tak, že obsahují ukazatele na všechny uzly, do kterých se lze z daného uzlu dostat, a také všechny ukazatele na předcházející uzly. Zmíněné ukazatele jsou uloženy ve strukturách pro uložení ukazatelů předchozích a následujících uzlů podle indexu, přičemž index představuje délku cesty od počátku do tohoto uzlu. Uzly

jsou z těchto struktur vybírány podle nejnižšího indexu. Podobně jako v Dijkstrově algoritmu jsou procházeny uzly podle nejmenší celkové vzdálenosti. Pomocí struktur přechozích a následujících uzlů jsou ukládány všechny možné cesty, přičemž aktivní je vždy pouze ta s nejnižším indexem – aktivní je tedy vždy pouze jeden následující uzel a jeden předchozí. V případě, že cesta obsahuje příliš mnoho uzlů, nebo jsme v cílovém uzlu, ale cesta obsahuje uzlů příliš málo, je volána funkce pro odstranění koncové části cesty, až po nalezení jiné alternativy.

Díky této modifikaci jsou uzly zkoumané vícekrát až v případě, kdy dojde k nalezení cesty nevhodné délky. Díky této modifikaci se podařilo snížit paměťovou i časovou náročnost algoritmu a průměrný čas zpracování jedná CAPTCHA se tím podařilo snížit na hodnoty kolem 0,2s.

Kapitola 6

Testování

V této kapitole si uvedeme způsob a výsledky dosažené při rozpoznávání reCAPTCHA a CAPTCHA API od Seznam.cz. Podíváme se také na příklady obrázků, u nichž bylo rozpoznání neúspěšné. Dále si také uvedeme možné způsoby rozpoznání jiných typů CAPTCHA.

6.1 reCAPTCHA

Nejlepších dosažených výsledků při rozpoznávání reCAPTCHA jsme dosáhli při použití následujících metod:

- Preprocessing: Grayscale, Invert, Remove Image, Power Values (exp=2), Auto Rotate (Angle from=-7, Angle to=7)
- Segmentation: Color Filling Segmentation, Nonlinear Histogram (Min Height=20, Min Width=50)
- Post-segmentation: Join Images (linear=1, min-max distance=24)
- Recognition: Rotate patern=22, Characters from=6, Characters to=8

S uvedeným nastavením bylo na testovací sadě obrázků reCAPTCHA dosaženo procentuální úspěšnosti 65,238% s počtem 137 správně rozpoznaných obrázků z celkového počtu 210 obrázků. Průměrná doba pro řešení jednoho obrázku byla 0,1s[1]. Dosažených výsledků vylo dosaženo s pouhými 5 ručně segmentovanými vzory od každého znaku.

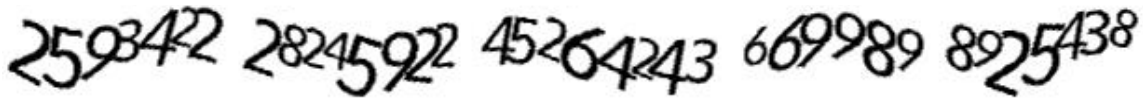
Schéma CAPTCHA se považuje za prolomené při úspěšnosti rozpoznání od 1% [2], tuto hranici se nám tedy podařilo mnohonásobně překonat. Díky vysoké rychlosti rozpoznání jsme schopni, během průměrné doby řešení jedné CAPTCHA člověkem (10s), správně automaticky vyřešit 65 těchto CAPTCHA. Těchto výsledků navíc dosahujeme i přes použití velmi jednoduchého klasifikátoru a extrakce vektoru rysů. Použitím pokročilejších klasifikátorů, jako neuronových sítí či SVM, bychom byli schopni dosáhnout výsledků ještě lepších.

Díky obrovskému rozšíření reCAPTCHA se tímto stávají tisíce služeb nezabezpečené, navíc je možné díky jejímu prolomení také automaticky zakládat e-mailové účty na serveru GMail.com, které se považují za důvěryhodnější a jsou tak méně často filtrovány jako spam. Jak bylo zmíněno v 3.1, počet zobrazení tohoto schématu je omezen, tento problém se však dá snadno obejít využitím botnetu, čímž se počet zobrazení rozloží přes velké množství počítačů. V případě prolomení způsobu kontroly počtu již zobrazených schémat tohoto

[1] Testování prováděno na stroji s procesorem Intel Core i5-4670K@3,8GHz

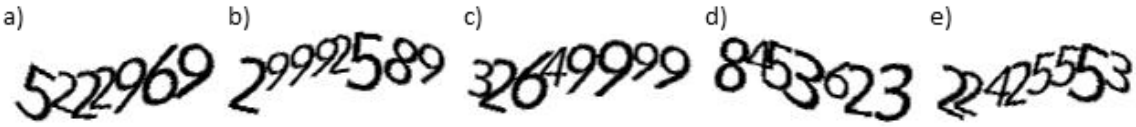
typu by pak byl jeden počítač schopný například registrace více než 23400 e-mailových účtů serveru GMail.com za 1 hodinu.

Na obrázku 6.1 vidíme příklady testovacích obrázků z reCAPTCHA, které byly rozpoznány správně, a to i přes použití překryvů a dotyků jednotlivých znaků.



Obrázek 6.1: Příklady správně rozpoznaných reCAPTCHA

Na obrázku 6.2 poté vidíme testovací obrázky, u nichž došlo k chybě při segmentaci znaků. Problém segmentace často způsobuje velký překryv jednotlivých znaků, obzvláště dvou znaků 2, které lze umístit velmi blízko sebe a jejich segmentaci tak velmi ztížit, což je vidět na případech a), e) na obrázku 6.2. Velký překryv způsobuje problémy segmentace také u jiných znaků, kde je využito změny jejich velikosti a vhodného umístění do volného místa jednoho ze znaků, což je možné vidět na případech b) znaky 9,2 a c) znaky 3,2 a 6,4 na obr. 6.2.



Obrázek 6.2: Příklady nerozpoznaných reCAPTCHA; problémové znaky: a) velký překryv znaků 2, 2; b) překryv znaků 9, 2; c) překryv znaků 3, 2 a 6, 4; d) téměř žádná mezera mezi znaky 4, 5; e) velký překryv znaků 2, 2

6.2 CAPTCHA API - Seznam.cz

Jako druhou testovací CAPTCHA jsme vybrali CAPTCHA API od společnosti Seznam.cz. Tato CAPTCHA je již starší a používá některé techniky, jako třeba statické pozadí, které v dnešní době nemají z pohledu zabezpečení žádný význam. Na rozdíl od mnohých jiných starších CAPTCHA umisťuje toto schéma znaky do obrázku značně pootočené a vertikálně i horizontálně posunuté. Díky tomu dochází v mnoha případech k dotyku znaků a v některých případech také k protínání znaků. Správná segmentace těchto protínajících se znaků je pak velmi obtížná.

Nejlepších výsledků jsme u tohoto schématu dosáhli při použití následujících metod

- Preprocessing: Grayscale, Pover Values (exp=0.2), Binarize, Invert, Remove Background (path=seznam_bg.png)
- Segmentation: Color Filling Segmentation, Nonlinear Histogram (Min Height=60, Min Width=30)
- Post-segmentation: Join Images (linear=1, min-max distance=35)
- Recognition: Rotate patern=35, Characters from=5, Characters to=5

S uvedeným nastavením bylo na testovací sadě čítající 62 obrázků správně rozpoznáno 50, čímž jsme dosáhli úspěšnosti 80,645%. Tohoto výsledku bylo navíc dosaženo při použití pouhého jednoho ručně segmentovaného obrázku pro každý znak. Průměrná doba řešení jednoho obrázku byla 0.1s.

Hranice prolomení CAPTCHA byla opět mnohonásobně překonána. Na rozdíl od reCAPTCHA má toto schéma větší množinu znaků, svoji bezpečnost si však značně narušuje statickým počtem znaků v obrázku (5), statickým pozadím a pouze náhodným dotykem a protínáním znaků.

Vzhledem k úspěšnosti a rychlosti rozpoznání se služby zabezpečené tímto typem CAPTCHA nedají vůbec považovat za zabezpečené.

Na obrázku 6.3 můžeme vidět, že i přes překrytí mohou být znaky segmentovány a rozpoznány správně, zvláště pak, pokud není jejich horizontální překrytí příliš velké, a pokud segmentací jednoho znaku nevzniká z druhého znaku jiný znak.

Obrázek 6.3: Správná segmentace u CAPTCHA API Seznam.cz

Naopak na obrázku 6.4 vidíme příklady obrázků, které byly rozpoznány špatně. Jednou z příčin vzniku špatné segmentace je takový překryv znaků, že segmentací jednoho znaku začně druhý znak pro klasifikátor vypadat jako znak jiný. Příkladem může být kombinace znaků UX (na obrázku 6.4a), kde po segmentaci znaku X vzniká ze znaku U znak C. Další příčinou špatné segmentace může být dotyk znaků mající velkou délku (obr. 6.4b). Řez poté není veden na předělu znaků, ale zkrz jeden ze znaků, čímž často vzniká jiný znak či dojde ke špatnému rozpoznání. Třetím případem, který špatnou segmentaci u tohoto typu CAPTCHA způsobuje, je velký horizontální přesah znaků (znaky H a B na obr. 6.4c), který způsobuje selhání použité metody segmentace, tedy nelineárního řezu. To je zde navíc posíleno protnutím znaků, čímž dochází k deformaci jednoho ze znaků a jeho špatnému rozpoznání.

Obrázek 6.4: Špatně segmentované CAPTCHA API od Seznam.cz; a) segmentace a rozpoznání UX jako CX; b) segmentace a rozpoznání OD jako CD; c) špatná segmentace HB

6.3 Další schémata

V této části se podíváme na možné způsoby předzpracování různých schémat CAPTCHA tak, aby byla možná jejich následná snadná segmentace a rozpoznání znaků. Samotnou segmentaci a rozpoznání již pro tato schémata provádět nebudeme. Jako vzorová schémata použijeme různá schémata komerčně nabízených CAPTCHA se serveru captcha.com, která představují zástupce různých typů schémat vyskytujících se na internetu.

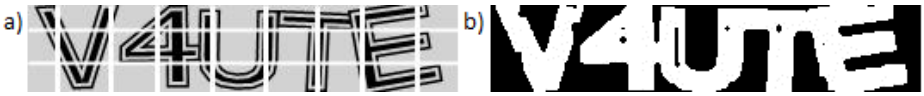
První zkoumané schéma s názvem Stitch je zástupcem schémat, které ve snaze o zabránění segmentaci "rozdrobí"znaky na malé čtverečky či obdélníčky (obr. 6.5a). Jsou-li však mezery mezi odbélníčky mezery o šířce 1px, je možné pro jejich vyplnění použít metodu předzpracování Fill 1px Gaps (obr. 6.5b). V případě i širších mezer mezi obdélníčky, je možné použít metodu Dilate, která rozšíří objekty do všech stran o 1px, čímž se mezery opět vyplní (obr. 6.5c).

Obrázek 6.5: Schéma Stitch s "rozdrobením"na obdélníčky; a) originální schéma; b) po zpracování metodou Fill 1px Gaps; c) zpracování metodou Dilate

Dalším příkladem je schéma s názvem CaughtInTheNet, které je zástupcem podobných schémat snažících se zabránit rozpoznání pomocí soustředných kružnic, elips či jiných tvarů z tenkých čar, které rozdělují znaky na části (obr. 6.6a). Zde stačí využít metod Fill 1px Gaps a metody Median s nastavenou velikostí matice zkoumaných prvků na 5. Výsledkem je vyplnění řezů od kružnic a následné odstranění tenkých čar mediánovým filtrem (obr. 6.6b).

Obrázek 6.6: Schéma CaughtInTheNets; a) originální schéma; b) po zpracování metodami Fill 1px Gaps a Median (size=5)

Třetím příkladem je schéma s názvem Split2, které je založeno na rozdělení obrázku pomocí horizontálních a vertikálních čar (obr. 6.7a). Opět zde můžeme využít metod Dilate a Fill 1px Gaps pro vyplnění mezer a tím připravení obrázku pro segmentaci (obr. 6.7b).



Obrázek 6.7: Schéma Split2; a) originální schéma; b) po zpracování metodami Dilate a Fill 1px Gaps

Jak je vidět, provedením několika vhodných metod předzpracování, po předchozím převodu do stupňů šedi a binarizaci, jsme jsme byli schopni převést různá schémata do stavu, kdy je možné provést segmentaci znaků a jejich rozpoznání. Server captcha.com samozřejmě nabízí schémat větší výběr, všechny by však šly snadno předzpracovat implementovanými, či na míru vytvořenými, metodami předzpracování – všechny se totiž zaměřují na zabránění rozpoznání znaků různými způsoby jejich vykreslení. Jednotlivé znaky jsou však téměř stejně velké a v obrázku pravidelně rozmístěné, což velmi usnadňuje odhad počtu znaků a jejich následnou segmentaci. Celková bezpečnost těchto schémat je tímto velmi oslabena.

6.4 Porovnání s jinými aplikacemi

Vývoj CAPTCHA je neustálý boj mezi těmi, kteří ji vyvíjí, a těmi, kteří se ji snaží prolomit. Porovnání výsledků s ostatními aplikacemi je velmi obtížné, jelikož s každým zveřejněným způsobem prolomení se schémata postižených CAPTCHA vyvíjí, čímž se opět stávají na nějakou dobu bezpečnými. Práce zabývající se tímto tématem vznikají často na akademické půdě a jejich cílem tedy není poskytnout útočníkům nástroj ke snadnému prolomení zabezpečení internetových služeb. Publikovány jsou tedy pouze dosažené výsledky těchto prací, samotné vytvořené aplikace již nikoliv. Naopak aplikace vytvářené útočníky se udržují v tajnosti, aby došlo ke změně napadené CAPTCHA co nejpozději a bylo tak možné nalezené zranitelnosti využívat co nejdéle. Kvůli tomu, že se obrázky CAPTCHA generují při každém jejich zobrazení, a navíc se neustále vyvíjí, neexistují také žádné standardizované sady testovacích dat sloužící k testování úspěšnosti aplikací při rozpoznávání CAPTCHA, jako je tomu například u strojového rozpoznávání textu či ručně psaného písma.

Jelikož nejsou dostupné ani jiné aplikace pro rozpoznání CAPTCHA, kromě komerčních, ani sady testovacích dat, je možné porovnat dosažené výsledky pouze v rovině čísel se zveřejněnými výsledky starších prací. V práci [6] v roce 2011 byla zveřejněna metoda segmentace pro reCAPTCHA a vlastní schéma CAPTCHA společnosti Google, u kterých byla dosažena úspěšnost rozpoznání 33%. V roce 2011 v práci [2] bylo pomocí vytvořené aplikace Decaptcha prolomeno 13 z 15 zkoumaných CAPTCHA, u kterých bylo dosaženo úspěšnosti od 2% do 93%. V práci [26] z roku 2008 byly zveřejněny nové techniky pro segmentaci znaků a proveden útok na různé typy CAPTCHA, včetně těch používaných společnostmi Microsoft, Yahoo! a Google. Úspěšnost segmentace u schématu MSN společnosti Microsoft byla 92% a celková úspěšnost rozpoznání tedy byla odhadnuta na více než 60%. Čím starší však práce je, tím jednodušší schémata CAPTCHA byla prolamována. I v případě zmiňovaných tří prací byly k zabezpečení CAPTCHA použity takové techniky, které by bylo možné z velké části odstranit již v rámci předzpracování.

Kapitola 7

Návrh pro zlepšení CAPTCHA

Tato práce potvrdila závěry předchozích prací (např. [2]), že bezpečnost CAPTCHA má být založena především na obtížné segmentaci znaků, a ostatní techniky, jako šum, přeškrtnutí čarami a další mají sloužit pouze k dalšímu posílení její bezpečnosti.

Uvedeme tedy seznam doporučení, které bychomm měli dodržet, chceme-li vytvořit bezpečnou textovou CAPTCHA:

Proměnlivý počet znaků Rozsah počtu znaků v obrázku by měl být dostatečně velký, například 5-9 znaků, což zhorší odhad počtu znaků a také sníží úspěšnost rozpoznáváním řízené segmentace.

Jednodušší sada znaků Není nutné volit příliš velkou sadu znaků. Je vhodné volit spíše menší počet možných znaků a vyhnout se znakům podobným (0-O, 1-7, i-j), které značně snižují úspěšnost rozpoznání u lidí.

Generování náhodných řetězců Generovaný řetězec by měl být vždy zcela náhodný, jinak je možné úspěšnost automatického rozpoznání značně navýšit použitím opravy výsledků podle slovníku

Různé typy písma Pro ztížení klasifikace znaků je vhodné používat dostatečné množství různých typů písem, čímž dojde ke snížení úspěšnosti klasifikace. Úspěšnost rozpoznání lidmi přitom snížena nebude.

Různé způsoby vykreslení znaků Jednotlivá schémata CAPTCHA používají vždy stejný způsob pro vykreslení všech znaků na obrázku. I přes to, že se mohou schémata v rámci jedné CAPTCHA měnit, je možné je podle určité vlastnosti identifikovat a použít vhodné metody předzpracování, čímž v obrázku ponecháme pouze zvýrazněné znaky určené k segmentaci. Použitím různých vykreslovacích metod pro jednotlivé znaky (např. tenké obrysy, rozdělení na malé obdélníčky, vyšší intenzita šumu) by bylo možné docílit stavu, kdy použitím jedné z metod předzpracování se některé znaky zvýrazní, jiné by však zanikly, čímž by se fáze předzpracování značně ztížila.

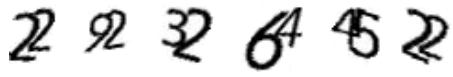
Rozdílná velikost znaků Pro zajištění účinnosti anti-segmentačních technik je nutností použít také různou velikost jednotlivých znaků, jinak je možné odhadnout počet a hranice znaků na základě určité vlastnosti, detekované například pouze u prvních dvou znaků.

Rotace Důležitou metodou k zabezpečení CAPTCHA je rotace jednotlivých znaků. Jelikož je možné klasifikátor naučit na různě pootočený vzor, neslouží rotace pro zabránění

procesu rozpoznání, avšak díky překryvům a dotykům znaků, které při rotaci vznikají, dochází ke ztížení procesu segmentace znaků.

Dotyk a překrytí znaků Pro vytvoření skutečně bezpečné CAPTCHA je nutné vykreslit znaky tak, aby se pokud možno zabánilo jejich správné segmentaci. Jako nejúčinnější se v tomto smeru jeví vykreslení znaků tak, aby se navzájem dotýkaly či překrývaly. Dotyk znaků nejen že zabraňuje použití Color Filling Segmentation, ale je-li větší, než tloušťka znaků v okolí dotyku, zabraňuje také segmentaci založené na řezech procházejících cestou s nejnižší celkovou intenzitou. Řez v takovém případě není veden v místě předělu mezi znaky, ale vede přes užší místo jednoho ze znaků, čímž jsou znaky deformovány pro následnou klasifikaci.

U překrývání znaků můžeme rozlišit dva způsoby - překrytí obdélníků vymezujících umístění znaků bez protnutí znaků a překrytí znaků s jejich protnutím. První způsob využívá prázdných oblastí znaků, do nichž je následně umístěn další znak. Pokud se znaky na alespoň na jednom místě dotýkají, není možné je oddělit pomocí CFS, a jejich segmentace je téměř nemožná, jelikož je těžké zjistit, zda se jedná o jeden či více znaků. Tento způsob je využit například u reCAPTCHA (příklady na obr. 7.1).



Obrázek 7.1: Překryv znaků bez jejich protínání

Druhou možností je umístit znaky tak, aby se navzájem protínaly. Obzvláště v kombinaci s rotací, různou velikostí a umístěním znaků je tato metoda velmi účinná. Vhodné je také vybrat a umístit znaky tak, aby jejich případnou segmentací, při ignorování protnutí znaků, vznikly znaky jiné. Ukázku použití této metody můžeme vidět u CAPTCHA API od Seznam.cz (obr. 7.2), kde však popisované protínání znaků nevzniká cíleně, ale pouze náhodně u některých obrázků.

Obrázek 7.2: Protínání znaků

Kapitola 8

Závěr

V rámci této práce jsme se seznámili s historií a vývojem CAPTCHA, jejími různými typy, problematikou jejího generování a také možnými způsoby jejího prolomení.

Jako hlavní typ CAPTCHA k prolomení byla vybrána reCAPTCHA, která patří mezi novější typy CAPTCHA, mající svoji bezpečnost založenou na antisegmentačních technikách, v tomto případě na překryvu a dotyku rotovaných znaků. Hlavním důvodem k jejímu výběru je její obrovské rozšíření a její úspěšné vyřešení více než 200 miliony uživateli každý den [1].

Provedli jsme analýzu tohoto schématu a navrhli řešení vedoucí k jeho prolomení. Navrhli a implementovali jsme tedy segmentační metodu, která kromě segmentace znaků reCAPTCHA umožňuje použití i u jiných sechémat. Tato metoda byla vytvořena kombinací několika segmentačních metod používaných při rozpoznávání textu a rozpoznávání starších typů CAPTCHA.

Za účelem testování vytvořené segmentační metody na různých schématech CAPTCHA byla navržena a implementována aplikace pro rozpoznávání CAPTCHA. Ta obsahuje různé metody pro jednotlivé fáze rozpoznávání CAPTCHA a umožňuje jejich libovolné kombinace pro dosažení co nejlepšího výsledku rozpoznání pro různá schémata CAPTCHA.

Účinnost vytvořené segmentační metody jsme otestovali na dvou schématech – reCAPTCHA a CAPTCHA API od Seznam.cz. U prvního zmíněného schématu jsme dosáhli úspěšnosti rozpoznávání 65,238% s časem 0,1s na rozpoznání jednoho obrázku, u druhého schématu pak 80,645% s časem 0,1s na obrázek. U obou těchto schémat jsme tedy několikanásobně překročili 1% úspěšnost, která se udává jako hranice pro považování schématu CAPTCHA za prolomené [2].

Na základě dosažených výsledků a získaných poznatků jsme uvedli doporučení pro návrh bezpečné CAPTCHA. Ta se týkají hlavně metod k zabránění segmentace znaků, což se považuje za hlavní mechanizmus zabezpečení CAPTCHA [2]. Jako nejúčinnější jsou doporučeny dotyk a překrytí znaků. Kromě toho jsou uvedeny také další způsoby, jak bezpečnost ještě posílit.

Úspěšnost rozpoznání by mohla být vylepšena použitím lepších klasifikátorů, založených například na neuronových sítích či SVM, a výběrem vhodné množiny příznaků pro klasifikátor. Lepších výsledků by také bylo možné dosáhnout implementací metody segmentace, která by umožňovala řez objektů v kterémkoliv místě a neomezovala se pouze na nelineární vertikální řezy, které mají omezené možnosti při velkém překryvu znaků. Vyšší rychlosti rozpoznání by mohlo být dosaženo také paralelním zpracováním obrázků ve více vláknech.

Literatura

[1] Ahn, L. v.; Blum, M.; Hopper, N.; aj.: The CAPTCHA Web page. www.captcha.net, 2000.

[2] Bursztein, E.; Martin, M.; Mitchell, J.: Text-based CAPTCHA strengths and weaknesses. In *Proceedings of the 18th ACM conference on Computer and communications security*, ACM, 2011, s. 125–138.

[3] Chellapilla, K.; Larson, K.; Simard, P. Y.; aj.: Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs). In *CEAS*, 2005.

[4] Dijkstra, E. W.: A note on two problems in connexion with graphs. *Numerische mathematik*, ročník 1, č. 1, 1959: s. 269–271.

[5] El Ahmad, A. S.; Yan, J.; Marshall, L.: The robustness of a new CAPTCHA. In *Proceedings of the Third European Workshop on System Security*, ACM, 2010, s. 36–41.

[6] El Ahmad, A. S.; Yan, J.; Tayara, M.: *The Robustness of Google CAPTCHA's*. Computing Science, Newcastle University, 2011.

[7] Lee, S.-W.; Lee, D.-J.; Park, H.-S.: A new methodology for gray-scale character segmentation and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, ročník 18, č. 10, Oct 1996: s. 1045–1050, ISSN 0162-8828, doi:10.1109/34.541415.

[8] Lillibridge, M.; Abadi, M.; Bharat, K.; aj.: Method for selectively restricting access to computer systems. Únor 27 2001, uS Patent 6,195,698.
URL http://www.google.com/patents/US6195698

[9] Mori, G.; Malik, J.: Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, ročník 1, IEEE, 2003, s. I–134.

[10] Moy, G.; Jones, N.; Harkless, C.; aj.: Distortion estimation techniques in solving visual CAPTCHAs. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, ročník 2, IEEE, 2004, s. II–23.

[11] Naor, M.: Verification of a human in the loop or Identification via the Turing Test. *Unpublished draft from http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human_abs.html*, 1996.

[12] Peterson, L. E.: K-nearest neighbor. *Scholarpedia*, ročník 4, č. 2, 2009: str. 1883.

[13] Pwntcha - captcha decoder web site: http://sam.zoy.org/pwntcha/.

[14] Sense for deafblind people: http://www.sense.org.uk.

[15] Simard, P.: Using machine learning to break visual human interaction proofs (hips. *Advances in neural information processing systems*, ročník 17, 2005: s. 265–272.

[16] SolveMedia: http://www.solvemedia.com.

[17] Turing, A. M.: Computing machinery and intelligence. *Mind*, ročník 59, č. 236, 1950: s. 433–460.

[18] Vicarious: http://www.vicarious.com.

[19] Von Ahn, L.; Blum, M.; Hopper, N. J.; aj.: CAPTCHA: Using hard AI problems for security. In *Advances in Cryptology-EUROCRYPT 2003*, Springer, 2003, s. 294–311.

[20] Wikipedia: Botnet — Wikipedia, The Free Encyclopedia. 2014, [Online; accessed 18-May-2014].
URL http://en.wikipedia.org/w/index.php?title=Botnet&oldid=605846766

[21] Wikipedia: Canny edge detector — Wikipedia, The Free Encyclopedia. 2014, [Online; accessed 18-May-2014].
URL http://en.wikipedia.org/w/index.php?title=Canny_edge_detector&oldid=606508539

[22] Wikipedia: Session ID — Wikipedia, The Free Encyclopedia. 2014, [Online; accessed 18-May-2014].
URL
http://en.wikipedia.org/w/index.php?title=Session_ID&oldid=596998971

[23] Wikislovník: bot — Wikislovník: Otevřený slovník. 2013, [Online; navštíveno 18. 05. 2014].
URL http://cs.wiktionary.org/w/index.php?title=bot&oldid=451647

[24] Xu, Y.; Reynaga, G.; Chiasson, S.; aj.: Security and usability challenges of moving-object CAPTCHAs: decoding codewords in motion. In *21st USENIX Security Symposium*, 2012.

[25] Yan, J.; El Ahmad, A. S.: Breaking visual captchas with naive pattern recognition algorithms. In *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual*, IEEE, 2007, s. 279–291.

[26] Yan, J.; El Ahmad, A. S.: A Low-cost Attack on a Microsoft CAPTCHA. In *Proceedings of the 15th ACM conference on Computer and communications security*, ACM, 2008, s. 543–554.

Příloha A

Obsah CD

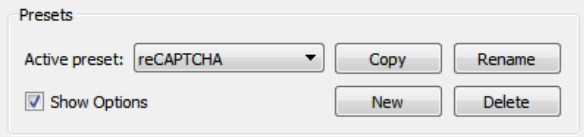
- bin_win32 – Složka obsahující aplikaci v binární podobě, včetně potřebných knihoven.
- data – Složka obsahující obrázky CAPTCHA pro testování aplikace.
- pisemna_zprava_src – Zdrojový tvar písemné práce.
- src – Zdrojové kódy aplikace.
- pisemna_zprava.pdf – Písemná zpráva diplomové práce.
- preklad.txt – Pokyny k překladu aplikace.
- prirucka.pdf – Uživatelská příručka aplikace.

Příloha B

Uživatelská příručka

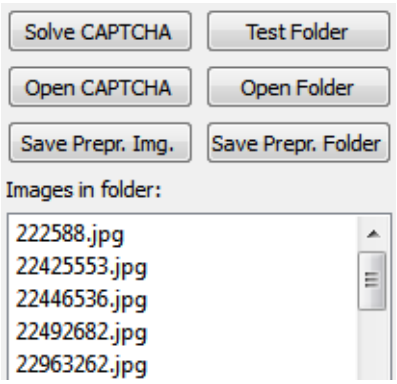
Nastavení aplikace pro rozpoznání CAPTCHA:

1. **Výběr presetu** – V levé horní části okna aplikace se nachází nastavení presetů (obr. B.1), tedy uložených nastavení pro jednotlivé typy CAPTCHA. To umožňuje výběr uloženého nastavení, vytvoření nového nastavení, kopii aktuálního nastavení, přejmenování a smazání nastavení.



Obrázek B.1: Nastavení presetu

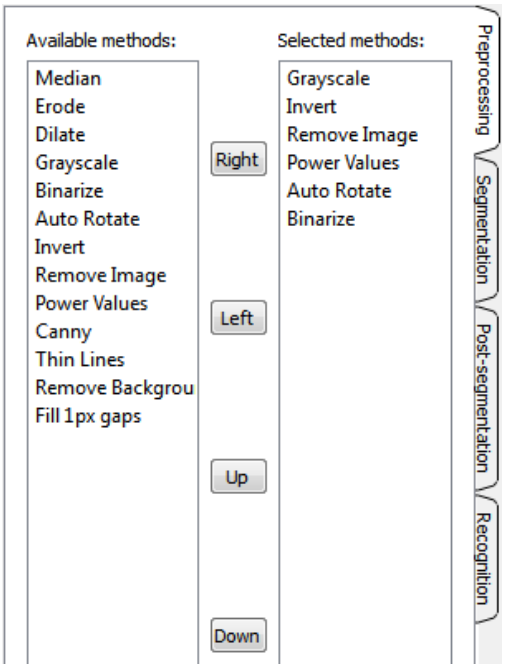
2. **Otevření obrázku CAPTCHA** – Výběr obrázku pro rozpoznání je možné provést tlačítkem Open CAPTCHA (obr. B.2), přetažením obrázku (Drag and Drop) do okna aplikace nebo výběrem názvu souboru v seznamu obrázků v aktuální složce, přičem aktuální složka se změní automaticky po otevření obrázku či výběrem složky pomocí tlačítka Open Folder.



Obrázek B.2: Otevírání a testování

3. **Předzpracování obrázku** – V nastavení (obr. B.3) v záložce Preprocessing vybereme vhodné metody k předzpracování obrázku. Dostupné metody jsou zobrazeny

v levém seznamu. Jejich výběrem a stiskem tlačítka Right dojde k jejich zařazení do aktuální nastavované fáze. Výběrem zařazené metody dojde k zobrazení jejích případných parametrů. Tlačítky Up a Down je možné měnit pořadí vybraných metod. Výsledek jedotlivých metod zkontrolujeme stiskem tlačítka Solve CAPTCHA (obr. B.2).



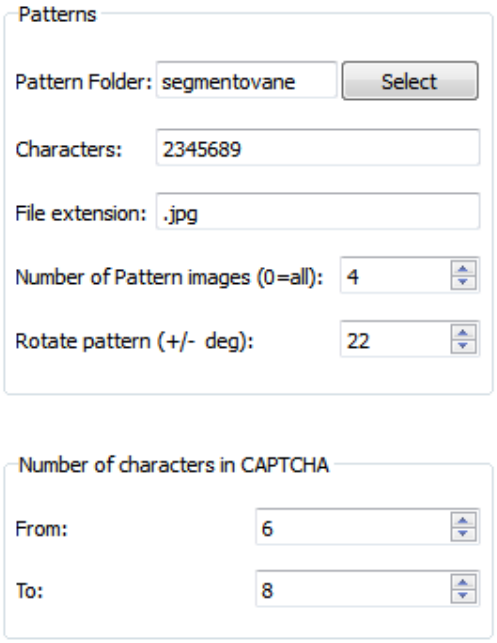
Obrázek B.3: Nastavení metod fází rozpoznání

4. **Příprava vzorů** – Po nalezení vhodných metod předzpracování je nutné vytvořit vzory jednotlivých znaků. Pro usnadnění práce je možné kliknout na tlačítko Save Prepr. Img. (obr. B.2), čímž dojde k uložení výsledného obrázku po předzpracování do podsložky aktuální složky s názvem preprocessed. Stiskem tlačítka Save Prepr. Folder dojde k uložení všech předzpracovaných obrázků z aktuální složky. Z těchto obrázků je následně nutné ručně vysegmentovat jednotlivé znaky a uložit je jako samostatné obrázky s názvy [znak]_[pořadí].[přípona], kde [znak] je znak obsažený v obrázku, [pořadí] je pořadové číslo vzorového obrázku (počítané od 1), a [přípona] je přípona souboru.

5. **Segmentace a Post-segmentace** – Následně je nutné, podobně jako v předzpracování, vybrat a nastavit metody pro fáze segmentace a post-segmentace.

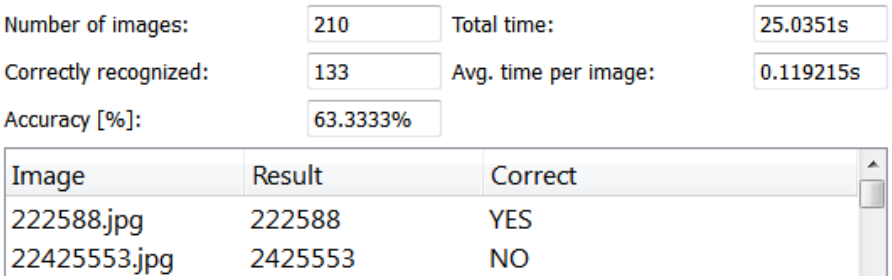
6. **Nastavení rozpoznávání** – Přepneme se na záložku Recognition a nastavíme parametry pro klasifikátor (obr. B.4). Vybereme složku obsahující vzorové obrázky, do pole Characters zadáme jednotlivé znaky obsažené v CAPTCHA, např. 123456789. Pro každý znak musí existovat alespoň jeden vzorový obrázek. Dále nastavíme příponu vzorových obrázků (včetně „.“, např. „.jpg“). Můžeme také omezit počet použitých vzorových obrázků nastavením hodnoty Number of Pattern images na jinou hodnotu, než 0. Pro rozpoznání rotovaných znaků je každý vzor vždy po jednom stupni pootočen a uložen jako vzor stejného znaku. Rozmezí pootáčení lze nastavit hodnotou

Rotate Pattern. Nakonec nastavíme minimální a maximální počet znaků, které může CAPTCHA obsahovat.



Obrázek B.4: Nastavení klasifikátoru

7. **Testování** – Rozpoznání aktuálního obrázku je možné provést stiskem tlačítka Solve CAPTCHA (obr. B.2). Přetažením obrázku do okna aplikace (Drag and Drop) nebo dvojklikem na název obrázku v seznamu obrázků v aktuální složce dojde k načtění a rozpoznání daného obrázku. Výsledky jednotlivých fází jsou zobrazeny jako obrázky v záložce Captcha Results. Pro hromadné testování vybereme pomocí Open Folder složku obsahující obrázky pro testování. Následně stiskem Test Folder provedeme testování všech obrázků z aktuální složky. Po zpracování všech obrázků je aktivována záložka Folder Results obsahující informace o testování (obr. B.5). Uveden je počet obrázků, počet správně rozpoznaných obrázků, procentuální úspěšnost rozpoznání, celkový čas zpracování obrázků a průměrný čas zpracování jednoho obrázku. Pod těmito informacemi je zobrazen seznam testovaných obrázků obsahující název obrázku, výsledek rozpoznání a informaci o správnosti rozpoznání. Správnost rozpoznání se provádí porovnáním výsledku s názvem souboru (bez přípony). Dvojklikem na název obrázku dojde k jeho rozpoznání a zobrazení výsledků jednotlivých fází zpracování.



Obrázek B.5: Výsledky testování složky