AUTONOMOUS CROSSWORD PUZZLES FOR ANDROID PLATFORM

Petr Buno

Bachelor Degree Programme (3), FIT BUT

E-mail: xbunop00@stud.fit.vutbr.cz

Supervised by: Vítězslav Beran

E-mail: beranv@fit.vutbr.cz

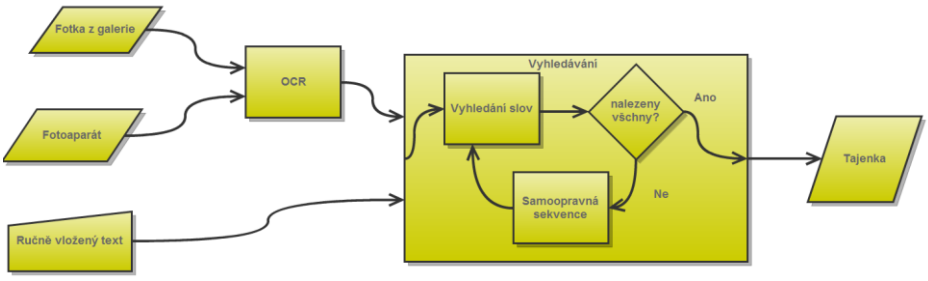
Abstract: This paper discusses the creation of aplication for autonomous crossword puzzles on Android platform. The goal is get the text from image and search words and than wanted solution from recognized text. This can be used to validate newly created or just solved crossword. For this purposes I created the prototype of aplication which uses Open Source OCR technology on Android platform.

Keywords: Crossword puzzle, Android, String search algorithm

1. ÚVOD

Tato práce pojednává o návrhu a vývoji metody pro automatické vyhledání tajenky osmisměrky. K ověření metody byla vytvořena aplikace na platformě Android používající technologie OCR.

Jedná se o jednoduchou aplikaci pro mobilní zařízení využívající vestavěný fotoaparát pro získání obrazových dat, uživateli je však umožněno vkládat data i v textové podobě. Cílem je vytvořit aplikaci, se kterou uživatel zvlášť vyfotografuje křížovku a hledaná slova a algoritmus za něj převede oba snímky na text. V takto získaném textovém vstupu křížovky nalezne všechna slova z rozpoznané množiny hledaných slov a zobrazí uživateli skrytou tajenku. Aplikace se musí vypořádat s nedokonalostí OCR enginu i se snímky ve špatné kvalitě.
Pro rozpoznávání vyfotografovaného textu je použita Open Source knihovna postavená na základech OCR enginu Tesseract, která obsahuje základní předzpracování obrazu, které má vliv na stabilitu převodu textu (více viz kap. 3).
Aplikace se specializuje na český typ křížovek čtvercového či obdélníkového tvaru, kde je cílem nalézt smysluplnou tajenku. Aplikace nedisponuje žádným druhem slovníku, a proto musí být zadána všechna vyhledávaná slova. Aplikace se umí vypořádat s českou, případně anglickou abecedou.



Obrázek 1: Schéma metody automatizovaného vyhledávání tajenky

2. AUTOMATICKÉ LUŠTĚNÍ OSMISMĚRKY

Cílem metody je nalézt tajenku osmisměrky z obrázku. K řešení je třeba dvou textových zdrojů - matice znaků reprezentující osmisměrku a seznam hledaných slov. Oba zdroje je nejdříve nutné nalézt ve vstupních snímcích a převézt na text.

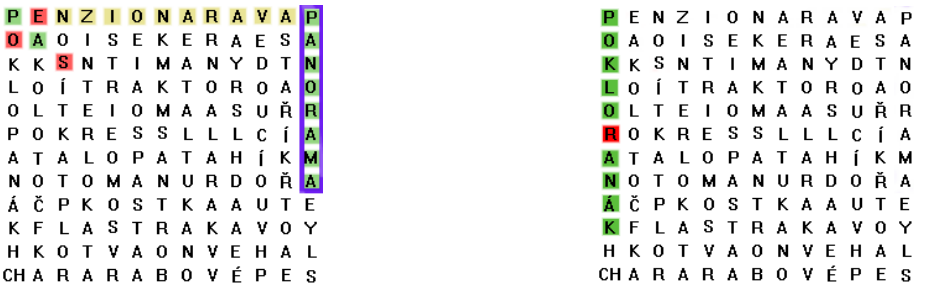
Rozpoznaný text je dále filtrován pro odstranění nepřesností. Pro vyhledávání slov byl vytvořen vyhledávací algoritmus, který vyhledá všechna slova a zobrazí skrytou tajenku. Dále je zde realizován samoopravný modul, který vyhledává slova i přes drobné chyby vzniklé nedokonalým rozeznáním a ty opravuje. Celý postup metody je vyjádřen pomocí blokového schématu na obrázku 1.

2.1. VYHLEDÁVACÍ ALGORITMY TEXTOVÝCH ŘETĚZCŮ

Na vyhledávání textových podřetězců existuje mnoho efektivních metod, jako je Boyer-Moorův nebo Knuth-Morris-Prattův algoritmus[1]. Tyto vysoce optimalizované metody však ztrácí účinnost ve spojitosti s nutností úpravy 2D textu pro jejich využití a následné značkování. Z toho důvodu jsem navrhl vlastní vyhledávací algoritmus, který je pro praktické využití zcela dostačující.

2.2. POPIS POUŽITÉHO VYHLEDÁVÁCÍHO ALGORITMU

Na obrázku 2 vlevo lze ukázat postup vyhledávacího algoritmu například na slově PANORAMA. Slovo se začíná vyhledávat od levého horního rohu. V případě, že je nalezeno první písmeno slova, je v Moorově okolí ve směru hodinových ručiček shora hledáno druhé písmeno. Pokud je nalezeno, je tím určen i směr, kde se potencionálně vyskytuje zbylá část slova (zde JV). Zvoleným směrem dále pokračuje rekurzivní procedura, která se v případě neúspěchu vrací k původnímu písmenu pomocí backtrackingu. Pokud je celé slovo nalezeno, jsou jeho písmena označkována, jako použitá a v další iteraci je hledáno následující ze seznamu slov. V případě nalezení všech slov je z neoznačených znaků sestavena tajenka.



Obrázek 2: Průchod vyhledávacího algoritmu, chybně rozpoznaný znak (R => P)

2.3. SAMOOPRAVNÝ MODUL

OCR enginy zatím nedokážou rozpoznat znaky vždy správně, především na fotografiích horší kvality, či při deformovaném textu. Z tohoto důvodu je nutné provést korekci jejich výstupu. Před samotným vyhledáváním je spuštěna automatizovaná korekce textu, která odhalí shluky nesmyslných znaků, případně jednoduché opravy typu „|“ (svislá čára) => „I“ (velké písmeno i).

Po prvním vyhledání je tento jednoduchý princip rozšířen o post-vyhledávací logiku, která se snaží zpětně opravit znaky v mřížce a v seznamu slov. Celý princip je založen na teorii, že při dostatečném množství hledaných slov lze s pomocí již nalezených slov validovat znaky v nich obsažené. Slova, která nebudou při prvním průchodu nalezena, budou hledána s možností použití dočasné transformace jednoho až n znaků na znak jiný. Lze předpokládat, že u delších slov bude úspěšnost vyšší, jelikož pravděpodobnost, že se stejná posloupnost netransformovaných znaků objevuje i jinde v mřížce, klesá s délkou hledaného slova.

Tento algoritmus zároveň ukládá transformované znaky do dynamické struktury, ze které se na konci vybere finální kandidát, který bude odpovídat nejvíce slovům křížících se na tomto místě.

Tento postup se opakuje, dokud je aktuální iterace úspěšná oproti té předchozí. V případě neúspěchu je vyžadována korekce ze strany uživatele.

Na obrázku 2 vpravo lze pozorovat záměnu P za R, což způsobilo chybu jak ve slově POKLOP tak ve slově PANÁK. Vzhledem k tomu, že je to v obou případech stejné písmeno a ani jedno slovo se jinde v mřížce nevyskytuje, je pravděpodobné, že byl znak špatně rozpoznán a je transformován na nejvhodnějšího kandidáta - P.

3. TESTOVÁNÍ

Testování probíhalo na několika sadách osmisměrek různých fontů a jazyků (český, anglický). Výsledkem testování bylo zjištění, že záleží především na kvalitě fotografie a úrovní deformování textu. Klíčovým prvkem k úspěšnému vyluštění je tedy ostrá a především selektovaná fotka. Na obrázku 3 je zobrazena jedna z testovacích sad s 100% úspěšností.

Obrázek 3: Ukázka reálných vstupních dat a jejich interpretace aplikací

I přesto, že budou nalezena všechna hledaná slova, což označí znaky jimi použité, zbudou zde nepoužité znaky, které tvoří skrytou tajenku. Z podstaty věci nedokáže postup popsaný v kapitole 2.3 odhalit chyby v těchto zbylých znacích a tajenka tak může obsahovat chyby. Vzhledem k tomu, že tajenka bývá často část věty či nějaké sousloví, má uživatel šanci, že se mu podaří tajenku doplnit. Pravděpodobnost že bude více písmen právě v tajence chybných, je nízká v případě úspěšného vyhledání všech slov. Na obrázku 3 vpravo lze naopak pozorovat, správně vyluštěnou tajenku i přes neprávně rozpoznaná slov (např. VOUSY => VOUSV).

4. ZÁVĚR

Výsledkem práce je navržení a realizace metody na automatické vyhledání tajenky v osmisměrce. Metoda je robustní vůči chybám. V rámci řešení jsem navrhl i GUI a celou aplikaci realizoval na platformě Android. Tento luštič je jednoduchou utilitou, která může luštitelům pomoci zkontrolovat jejich dosažené výsledky.

PODĚKOVÁNÍ

Tento příspěvek vznikl za podpory grantu FIT-S-11-2 a výzkumného záměru MSM 0021630528.

REFERENCE

[1] BOYER, Robert S. a J. Strother MOORE. A fast string search algorithm. *A fast string search algorithm* [online]. 1977, č. 20 [cit. 2013-03-25]. Dostupné z: http://www.cs.utexas.edu/~moore/publications/fstrpos.pdf