мр Велибор Илић,
ilicv@EUnet.yu,
http://solair.EUnet.yu/~ilicv/

ОЦР – програм за препознавање ћириличних слова

Абстракт**: У раду се говори о програму ОЦР за препознавање ћириличних слова. Описани су проблеми који се јављају приликом израде софтвера за препознавање слова и предложена су нека од решења која се могу применити да би се превазишли постојећи проблеми. Описан је и поступак примене неуронских мрежа у препознавању слова.**

Кључне речи**: ОЦР, препознавање ћириличних слова, препознавање облика, неуронске мреже, објектно оријентисано програмирање, Delphi**

Увод

Проблем препознавања облика је људима веома близак; са њиме се срећу током целог живота, почевши од најранијег детињства. Препознавање облика није урођена особина него се стиче учењем. Способност препознавања облика људи развијају и усавршавају готово целог свог живота. До недавно су само живи организми имали способност препознавања облика. У последње време, захваљујући развоју информационих технологија и вештачке интелигенције, остварени су конкретни резултати у области препознавања облика. Под појмом теорије препознавања облика подразумевамо математичке методе, првенствено намењене аутоматском класификовању (машинском) конкретних објеката. Теорија препознавања облика ослања се углавном на употребу метода вештачке интелигенције као што су: неуронске мреже, фази логика, експертни системи, математичка логика, статистичке методе, итд.

Проблем препознавања скенираног текста се може свести на проблем препознавања слова која су саставни делови текста. За методе препознавања слова се на енглеском језику користи термин “Optical Character Recognition” (OCR).

Проблем препознавања текста спада у проблеме за које рачунари нису најбоље прилагођени. Тежина проблема је у тешкоћи да се дефинише скуп правила помоћу којих би описали мноштво варијација слова. Једно исто слово се у неком тексту може појавити у више различитих величина и у много облика (фонтова). Са оваквим варијацијама човек лако излази на крај, али рачунари имају озбиљне потешкоће.

Слова се разликују према:

- фонтовима (Arial, `Courier`, Times New Roman, …)
- форматирању (Нормал, *Италик*, **Болд**, ***Болд италик***)
- ВЕЛИКА СЛОВА, мала слова
- величини фонта у пикселима (величина 8, величина 14, величина 20, …).

Проблем препознавања слова се може посматрати као проблем класификације скупа облика (слова) на међусобно дисјунктне класе. Сваки посматрани објекат (облик) карактерише скуп његових особина на основу којих се дати објекат разликује од осталих. За теорију препознавања облика нису интересантне све особине објекта, него само оне особине на основу којих се дати објекат разликује од осталих, тј. особине по којима се може класификовати.

Када не би долазило до оштећења слова приликом штампања и касније приликом скенирања, овај поступак би се могао решити тополошки. Слова би се прво класификовала према карактеристичним особинама као што су: број пресека линија, број затворених линија, број крајева итд. Слова би смо прво сврстали у класе сличних слова. На пример, слова Т и Ү би чинила једну класу слова, К и Х другу, У и В, трећу, А и Р четврту и слично. У другој етапи би се слова квалитативно анализирала у оквиру исте класа. Овај концепт је тешко применљив из поменутог разлога што приликом штампања и скенирања долази до оштећења слова.

Препознавање слова најчешће се врши помоћу: неуронских мрежа, експертних система или Бајесових метода.

Принцип препознавања ћириличних слова је веома сличан општем проблему препознавања слова. Разлог због чега се страни програми за препознавање текста не могу применити за препознавање ћириличних слова је то што у скупу латиничних слова не постоје нека ћирилична слова као што су (Б, Г, Д, Ђ, Ж, З, И, Л, Љ, Н, Њ, П, Ћ, У, Ф, Ц, Ч, Ш).

Софтвер за препознавање текста

Софтвер за препознавање текста се користи да би се избегло прекуцавање текста који се налази на папирним документима ради касније обраде, корекције или меморисања у електронском облику. На тржишту се може наћи софтвер за препознавање текста као што је “Recognita Plus”, TextBridge и OmniPage. Недостатак ових програма је у томе што су углавном намењени за препознавање латиничних слова.

Један од најпознатих и најраспрострањенијих програма за препознавање теста је програм Recognita Plus. Recognita има подршку за препознавање YU латиничних слова (ŠĆČĆŽ). Препознавање текста код овог програма се базира на контурној анализи и допуњена је *бит-мацтхинг*-ом (поређење текућег карактера са унапред припремљеним идеалним словима).

На нашим просторима је средином деведесетих, на Електротехничком факултету у Београду (1996), развијен софтвер под именом ИЦРА за препознавање ћириличних слова. Само препознавање слова се вршило помоћу теорије графова, а проблеми препознавања оштећених слова су решавани применом експертних система, односно сложеним стаблом одлучивања.

Проблеми препознавања слова

Приликом препознавања слова карактеристични су следећи проблеми:

- Слова различите величине
- Иста слова различитог изгледа
- Проблем спојених слова
- Проблем оштећених слова
- Проблем одређивања краја речи

Слова различите величине

У тексту се често јављају слова различите величине. На пример, наслови су већи од осталог текста, а и сам текст не мора бити увек исте величине. Човек без проблема може да чита слова различите величине, док је рачунару неопходно да улаз буде стандардизован тј. константне величине, за коју је обучавана неуронска мрежа. Проблем различитих величина слова се решава скалирањем издвојених слова.

Иста слова различитог изгледа

Због коришћења различитих типова слова (фонтова), често се дешава да иста слова изгледају другачије. Код неких слова велика и мала слова изгледају другачије (А,а). Такође код неких фонтова мала слова не изгледају исто у свим фонтовима (слика 1).

А **А** *А* А **А** *А* А **А** *А*	Г **Г** *Г* Г **Г** *Г* Г **Г** *Г*
а **а** *а* а **а** *а* а **а** *а*	г **г** *г* г **г** *г* г **г** *г*
Т **Т** *Т* Т **Т** *Т* Т **Т** *Т*	Д **Д** *Д* Д **Д** *Д* Д **Д** *Д*
т **т** *т* т **т** *т* т **т** *т*	д **д** *д* д **д** *д* д **д** *д*

Слика 1 *Иста слова различитог изгледа*

Проблем спојених слова

Први корак у препознавању текста је издвајање појединачних слова из текста. Уколико се слова додирују постоји проблем како разграничити где се завршава једно, а где почиње друго слово. Ово се често дешава код серифних фонтова као што је на пример Times New Roman. На слици 2 су означена спојена слова. Понекад се овај проблем може решити новим скенирањем текста у већој резолуцији.

Слика 2 *Спојена слова*

Укошена слова такође могу представљати проблем за издвајање појединачних слова из теста (слика 3).

Слика 3 *Укошена слова*

Проблем оштећених слова

Овај проблем се најчешће јавља при скенирању текста штампаног на лошијим штампачима или куцаним на писаћој машини.

На слици 4 је приказано оштећено слово Б. Оштећена слова се теже препознају јер се разликују од идеалних модела слова. Алгоритам за препознавање мора бити флексибилан да би могао да препозна оштећена слова. Из тог разлога се користе неуронске мреже јер оне поседују способност да препознају облике са одређеним степеном оштећења (шума).

Слика 4 *Делимично оштећено слово Б*

Много већи проблем представља прекидање слова на кључном месту, као што је приказано на слици 5. Велика је вероватноћа да ће програм за препознавање текста прекинуто слово препознати као два знака. На слици 5. је приказано прекинуто слово П.

Слика 5 *Прекинуто слово П*

Проблем одређивања краја речи

Приликом препознавања слова се јавља проблем раздвајања речи. Како одредити где је почетак, а где крај једне речи. Речи се одвајају тако што се између њих налази празан знак. Размак између слова може бити различите дужине што ствара проблем. Такође постоје пропорцијални и непропорцијални фонтови код којих је растојање између слова различито.

Проблеми

Слика 6 *Пропорционални фонт*

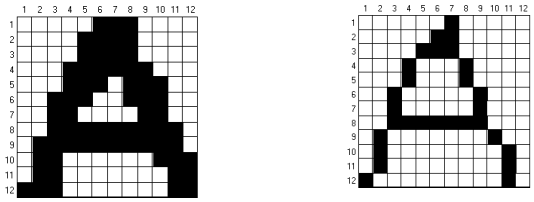
Проблеми

Слика 7 *Непропорционални фонт*

На слици 7, у случају непропорционалног фонта, се види да је растојање између слова л и е веће него код пропорционалног фонта (слика 6).

Стањивање слова

Стањивање се састоји у уклањању сувишних пиксела, а да при томе слово задржи препознатљив облик. У неким радовима се помиње да се слова успешније препознају уколико се ивице слова стање на дебљину од једног пиксела. Теоретски, стањивање слова би требало да смањи потребан број примера за обучавање мреже.



Слика 8 *Слово пре и после стањивања*

Приликом стањивања слова долази до губитка информација и то је нарочито проблематично код оштећених слова или код нејасних слова. На тај начин би се код нејасних слова нејаснођа још више повећала.

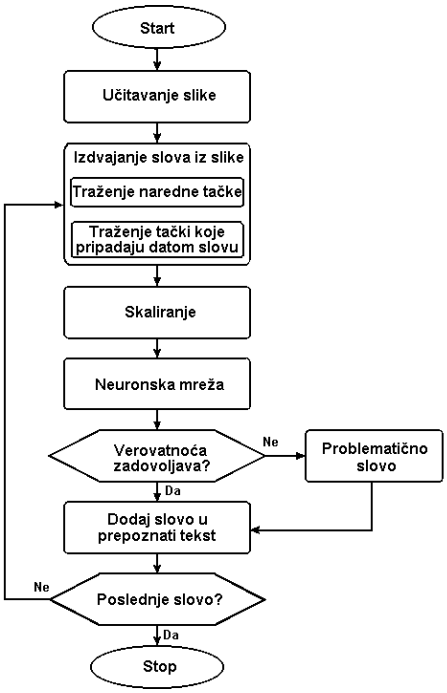
На слици 8 се види слово пре и после процедуре стањивања. У пракси се, током тестирања програма “ОЦР”, показало да се приликом стањивања губи део информација, па би слово након процедуре стањивања постало прилично нечитљиво. Из наведених разлога се код овог софтвера одустало од примене процедуре стањивања.

ОЦР - Програм за препознавање текста

Алгоритам софтвера ОЦР за препознавање текста

У даљем тексту је дат кратак опис алгоритама по коме ради “ОЦР” програм за препознавање текста.

Учитавање слике представља први корак у алгоритму препознавања. Са диска се учитава унапред припремљена слика у битмап формату.



Слика 9 *Алгоритам за препознавање слова*

Издвајање слова из слике – да би могли да препознамо слово, морамо га издвојити од остатка слике. Слова се издвајају једно по једно, одговарајућим редоследом (слева на десно и од горе према доле), на исти начин као што то чини човек који чита текст. Издвајање слова из слике је подељено у две операције.

Тражење једне од тачка наредног слова. Ова процедура анализира слику и проналази координате тачке која припада следећем слову. Приликом стартовања програма претрага се почиње од горњег левог угла слике (0,0).

Одређивање најмањег правоугаоника у коме се налази комплетно слово. Да би смо могли издвојити појединачна слова из скениране слике потребно је утврдити које све тачке припадају датом слову. Када се претходном процедуром добију координате једне од тачака које припадају датом слову, потребно је одредити најмањи правоугаоник који обухвата све тачке датог слова.

У овом програму се најмањи правоугаоник одређује методом концентричних кругова. Око нађене тачке се испитују тачке које окружују нађену тачку. Затим се кругови повећавају док се не добију необојене тачке са све четири стране. Недостатак ове методе је што не може да раздвоји слепљена слова а проблеми се могу јавити и код укошених слова, као што приказано на слици 2.

Скалирање се врши да би програм био независан од величине слова. Скалирањем се сва слова, без обзира на величину, своде на матрицу 12×12 каква је коришћена приликом обучавања неуронске мреже.

Довођењем слова стандардне величине на улаз **неуронске мреже,** на излазу се добија тридесет реалних бројева у интервалу [0,1] који представљају вероватноћу слова са улаза. Одабира се слово са највећом вероватноћом. Уколико је слово на улазу коректно препознато, на излазу би требало да се појави низ од тридесет бројева; при чему би један од излаза имао вредност један, док би остали имали вредност приближно нула. Редни број излаза који има вредност један представља редни број слова у азбуци. У програму “ОЦР” неуронску мрежу можемо посматрати као црну кутију (непознату сложену логичку функцију).

У наредном кораку се испитује да ли је **вероватноћа** препознавања слова већа од границе толеранције. Уколико јесте, **слово се додаје препознатим словима**, а уколико није појављује се екран на коме се приказује **проблематично слово** и омогућава кориснику да покуша да препозна слово и ручно упише “проблематично” слово, које се затим придружује осталим препознатим словима.

Поступак се понавља све док се не дође до последњег слова на слици.

Првобитно је замишљено да се издвојено слово после скалирања **стањи** (пре пропуштања кроз неуронску мрежу), да ивице слова буду дебљине од једног пиксела, али се касније одустало од примене процедуре стањивања, јер се стањивањем губи важан део информација које је слово носило пре стањивања слике (слика 8).

Конфигурација неуронске мреже у програму за препознавање ћириличних слова

Програм за препознавање ћириличних слова “ОЦР” користи трослојну неуронску мрежу која има следећу конфигурацију:

- Трослојна неуронска мрежа са бацкпропагатион алгоритмом обучавања
- Број неурона на улазном слоју (број улаза): 12×12 = 144
- Број неурона на средњем слоју: 35
- Број неурона на излазном слоју (број излаза): 30
- Број примера којима се мрежа обучава: 1590 (30 слова у више варијанти)

Тренинг скуп којим је обучавана неуронска мрежа састојао се од 1590 слова. За обучавање ове мреже коришћени су фонтови: Ариал, Тимес Нew Роман и Цоуриер Нew. Слова су писана у облицима нормално (Нормал), укошено (Италиц), и подебљано (Болд). Такође су употребљене три величине фонта које су затим скалирањем сведене на величину 12×12. За све типове слова су коришћена велика и мала слова.

Време потребно за обучавање мреже износи 02:12:53 сати без прекида на рачунару Целерон 333. Да би се мрежа обучила, било је потребно 667 итерација. Максимална грешка је износила 0.098874 а просечна грешка је износила 0.00573.

ОЦР – софтвер за препознавање ћириличних слова

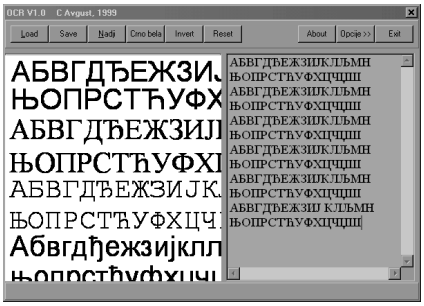
Програм “ОЦР” је написан у Borland Delphi 5 развојном окружењу.

Минимална хардверска конфигурација потребна за употребу програма за препознавања текста представља “PC” рачунар базиран на процесору класе “Intel Pentium” 100 или јачем, са 16 или више мегабајта RAM меморије и хард диск од око 5MB слободног простора (програм за препознавање текста заузима 1.5MB простора на хард диску, а остали простор је потребан за смештање слика које се обрађују).

Програм захтева да на рачунару буде инсталиран оперативни систем Microsoft Windows 95 или новији.

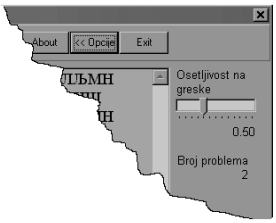
Кориснички интерфејс видимо на слици 10. Командни тастери у горњем делу екрана имају следеће функције:

- Лоад – учитава скенирану битмап слику са диска. Учитана слика се приказује у левој половини екрана.
- Саве – снима препознати текст са десне половине дела прозора у фајл.
- Нађи – команда којом се покреће препознавање текста.
- Црно бела – конвертује слику у црно белу верзију. Покретањем ове команде се елиминишу нијансе сиве боје. Ова опција омогућава прецизније препознавање слова.
- Инверт – команда за инвертовање слике. Служи за добијање негатива слике.
- Ресет – брише препознати текст из десне половине екрана и омогућава поновно препознавање. Ова опција се користи пре стартовања поновног препознавања текста након подешавања осетљивости на грешке.
- Абоут – информације о програму.
- Опције – одабирањем овог тастера појављује се са десне стране екрана додатни мени (слика 11.) који омогућава додатно подешавање програма.
- Ехит – напуштање програма.



Слика 10 *Кориснички Интерфејс програма*

Програм дозвољава промену прага осетљивости на грешке приликом препознавања текста. Иницијална вредност је 0.50, али се осетљивост на грешке може мењати у интервалу од 0.30 до 0.90. Уколико је вредност овог параметра већа, програм ће тачније препознавати слова, али ће зато чешће пријављивати “проблематична” слова.



Слика 11 *Подешавање осетљивости на грешке*

Уколико програм приликом препознавања текста наиђе на слово код кога је вероватноћа слова мања од жељене, препознавање се зауставља и појављује се дијалог (слика 12) на коме корисник може ручно унети “проблематично” слово. Програм у горњем десном углу приказује увећану слику проблематичног слова и предлаже слово са највећом вероватноћом. Вероватноћу предложеног слова можемо видети у горњем десном углу екрана. Слово које програм предложи веће је од осталих слова која се налазе на приказаним тастерима у левој половини екрана. Уколико је програм добро претпоставио слово, довољно је притиснути дугме “Ок” и програм ће слово убацити у препознати текст, а затим наставити са препознавањем. Уколико је програм погрешно препознао слово, корисник може помоћу миша да одабере то “проблематично” слово. Ако корисник жели, може “проблематично” слово да меморише и касније да покрене програм за обучавање. Након обучавања, мрежа ће успешно препознавати и таква слова. У десном делу екрана, испод слике се налази поље у коме се приказује последњи ред препознатог текста да би корисник могао лакше идентификовати слово у зависности од претходних слова.

Слика 12 *Слово са мањом вероватноћом од граничне*

Најчешћи разлог због чега програм зауставља препознавање је уколико наиђе на спојена слова (слова која нису правилно издвојена са слике). Таква "слова" програм не може да обради пошто је предвиђена обрада само појединачних слова.

Таква "слова" програм лако уочава јер имају малу вероватноћу. У том случају је потребно из леве половине прозора мишем одабрати оба слова која су приказана у горњем десном делу екрана.

Слика 13 *Спојена слова*

Препознати текст можемо меморисати на диску одабирањем опције **Сними**. Након одабирања опције Сними, појављује се дијалог у којим се одређује име фајла под којим желимо да текст снимимо на диск.

Закључак

Програми за препознавање текста су значајни из разлога што постоји велики број текстова (информација, знања) који су записани на папиру (књиге, пословни документи, фактуре, признанице, рачуни, итд.). Пребацивање ових информација у дигитални облик захтева велики људски рад. Пребацивањем информација записаних на папиру у дигитални облик, остварују се бројне предности као што су:

- лако претраживање података,
- лако умножавање,
- лако преношење,
- једноставну модификацију,
- уштеду простора (на малом простору се могу сместити велике количине информација),
- рачунске операције над подацима (уколико су у питању бројчане вредности).

Данашњи програми за препознавање текста, иако имају високу тачност приликом препознавања текста (преко 99%), још увек не могу да гарантују 100% сигурност приликом конвертовања информација. Када се постигне већа сигурност конвертоване информације, програми за препознавање текста ће наћи бројне примене у привреди и свакодневном животу.

Програм за препознавање ћириличних слова "ОЦР" је израђен у експерименталне сврхе током израде магистарског рада "Обучавање неуронских мрежа за препознавање ћириличних слова". Уколико се појави интересовање, овај софтвер би се могао прерадити у праву комерцијалну апликацију ако би се унапредили поједини модули програма и додале нове функције. Програм би функционисао много ефикасније када би се усавршио алгоритам за издвајање слова из слике. Ова верзија програма не разликује велика и мала слова, што би у комерцијалној верзији требало исправити. Постојећи програм би се могао веома лако прерадити за препознавање латиничних слова или неких других специфичних знакова/симбола. Програм за препознавање текста би се могао побољшати и интегрисаним системом за исправљање неправилних речи (спелл цхецкер) и, евентуално, системом за граматичку проверу речи у реченици. И поред ових наведених недостатака, реализовани програм показује добре резултате при препознавању текста.

Демо верзија ОЦР софтвера се може слободно преузети са Интернета на локацији:

Литература

[1] Илић, В., (1999) “Обучавање неуронских мрежа за препознавање ћириличних слова”, магистарски рад, Технички Факултет “Михајло Пупин”, Зрењанин

[2] Илић, В. (2000), “ NeuroVCL components for Delphi ”, НЕУРЕЛ 2000, Завод за графичку обраду Технолошко-металуршког факултета, Београд

[3] Илић, В. (2000), “ Force learn algorithm – training neural networks with patterns which have highest errors ”, НЕУРЕЛ 2000, Завод за графичку обраду Технолошко-металуршког факултета, Београд

[4] Берберски, Ж., (1996).: “Шта све могу неуронске мреже?”, часопис “Рачунари” бр 120, БИГЗ, Београд

[5] Bernander, O., (1998) “Neural Network”, Microsoft(R) Encarta(R) 98 Encyclopedia. (c) 1993-1997 Microsoft Corporation

[6] Хотомски, П., (1995): “Системи Вештачке интелигенције”, Технички факултет “Михајло Пупин”, Зрењанин

[7] Јоцковић, М., Огњановић З., Станковски С. (1997) “Вештачка интелигенција интелигентне машине и системи”, Графомед, Београд

[8] Миленковић, С., (1997): “Вештачке неуронске мреже”, Задужбина Андрејевић, Београд

[9] Никић, Б., (1990): “Примена теорије препознавања облика код обраде отиска папиларних линија прстију”, магистарски рад, ТФ “Михајло Пупин”, Зрењанин

[10] Николић, Т., Опачић, М., (1995): “Вештачка интелигенција и неуронске мреже”, ИБН Центар, Београд

[11] Ристановић, Д., (1994): “Читање… како то тешко звучи”, часопис “Рачунари” бр 103, БИГЗ, Београд

[12] Реисдорпх, К., (1998): “Научите Делфи за 21 дан”, Компјутер библиотека, Београд

[13] Сајић, И., (1995): “Неуронске мреже”, часопис “Рачунари” бр 108, БИГЗ, Београд

[14] Сајић, И., (1996): “Да ли је ПЦ научио да чита”, часопис “Рачунари” бр 119, БИГЗ, Београд

[15] Субашић, П., (1998): “Фази логика и неуронске мреже”, Техничка Књига, Београд

[16] “Frequently asked questions about AI”, http://www.cs.cmu.edu/Web/Groups/AI/html/faqs/ai/ai_general/top.html

[17] “Neural Network Frequently asked questions”, ftp://ftp.sas.com/pub/neural/FAQ.html