

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
MATEMATINĖS INFORMATIKOS KATEDRA

**Vytautas Jankauskas** *parašas*  
Bioinformatikos studijų programa

**Lietuviško rankraščio atpažinimas ir taikymas**  
Bakalauro baigiamasis darbas

Vadovas: lektorius **Irus Grinis** *parašas*

Vilnius 2017

# Turinys

<b>Įvadas</b>	<b>3</b>
<b>1 Kompiuterinės regos biblioteka OpenCV</b>	<b>6</b>
1.1 A1 pavadinimas . . . . .	9
1.2 A2 pavadinimas . . . . .	9
<b>2 Problemos sprendimas</b>	<b>9</b>
2.1 Pradinis vaizdo apdorojimas . . . . .	9
2.2 B2 pavadinimas . . . . .	9
2.2.1 B21 pavadinimas . . . . .	9
2.2.2 B22 pavadinimas . . . . .	10
2.3 B3 pavadinimas . . . . .	10
<b>Literatūra</b>	<b>10</b>
<b>Santrauka</b>	<b>11</b>
<b>Summary</b>	<b>11</b>

## Įvadas

Kompiuteriu spausdinta ir ekrane matoma tekstinė medžiaga po truputį keičia žmonių rašymo ir skaitymo kultūrą bei yra naudojama vis plačiau. Tačiau ranka parašytas tekstas vis dar yra (ir tikriausiai visada bus) neatsiejama žmonių gyvenimo dalis. Tai ypač jaučiama švietimo įstaigose – mokyklose, kolegijose, universitetuose, kur svarbesnė informacija užrašoma ranka, kad būtų labiau išryškinta ir geriau įsiminta.

Žmonėms rega atrodo savaime suprantamas ir elementarus jutimo būdas. Jau ankstyvoje vaikystėje pradedama sąmoningai atpažinti įvairias spalvas, formas, objektus. Akimis matomas vaizdas smegenyse išskirstomas į įvairius signalus kuriais perduodama skirtingų tipų informacija. Pavyzdžiui, norint aptikti aplinkoje konkretų objektą, smegenys analizuoja tik svarbesnes, turinčias reikiamas savybes, matomo vaizdo dalis. Žmogus per savo gyvenimą apdoroja labai daug informacijos gaunamos visais jutimo organais. Ta informacija leidžia smegenims sukurti be galo daug įvairių sąryšių, padedančių atpažinti objektus.

Kompiuterinis vaizdinės medžiagos apdorojimas išsikulto tikslo įvykdymui yra vadinamas kompiuterine rega [1]. Ji iš esmės turi tokią pat paskirtį kaip ir žmogaus rega. Tačiau kompiuteriai, skirtingai nei žmonės, vaizdinę medžiagą supranta tik kaip ilgą skaičių seką. Dirbtinės regos sistemos neturi jokių objektų atpažinimo modelio, nežino kurioje vietoje reikėtų fokusuoti vaizdą ar kurias jo dalis ignoruoti. Taip pat jos pačios neturi jokių sąryšių sistemų, kurios leistų palengvinti užduočių vykdymą. Taip pat bet kokiame vaizde egzistuoja įvairus triukšmas iš aplinkos, kurį sukuria kintantis apšvietimas, oro sąlygos.

Kompiuteriai visus su rega susijusius uždavinius gali išspręsti naudodami tik vaizdinę informaciją ir papildomą informaciją, kurią jiems suteikia žmonės tų užduočių įgyvendinimui. Kompiuterinė rega itin naudinga sprendžiant problemas, kurios iš žmonių reikalauja daug laiko, yra pasikartojančios. Todėl, nepaisant įvairių sunkumų, jos taikymų atsiranda vis daugiau:

- Pastaraisiais metais vis daugėja automatinio automobilių identifikacijos numerių atpažinimo sistemų. Kuriant tokias sistemas atsižvelgiama į įvairius veiksnius – numerių lokaciją, skaičių, kiekį, švarumą (lentelė su numeriais gali būti iš dalies padengta dulėmis ar purvu), taip pat į apšvietimo sąlygas, vaizdo foną [3].



Pav. 1: Supaprastinta automobilių numerių atpažinimo schema [7].

- Vienas iš seniausių kompiuterinės regos taikymų yra pašto indeksų atpažinimas ant vokų ir siuntinių, leidžiantis sutaupyti daug laiko paštų darbuotojams [8].



Pav. 2: Pašto indeksų pavyzdžiai iš Kinijos pašto.

- Beveik kiekvieno išmaniojo telefono kameroje yra galimybė naudoti veido aptikimo funkciją.
- Vietose, kur renkasi dideli pėsčiųjų srautai, skaičiuojamas praeinančių žmonių kiekis. Tam naudojamos įprastos vaizdo kameros vaizdo srautą segmentuojant ir jame išskiriant kiekvieną žmogų atskirai [5].
- Sukurtas maliarijos diagnozavimo metodas, naudojant kompiuterinę regą [2]. Gydytojui leidžia daug efektyviau dirbti, kadangi įrankis analizuoja kraujo mėginių atvaizdus ir atrenka tik tuos, kuriuose didžiausia maliarijos infekcijos tikimybė.

Šie pavyzdžiai rodo, kad kompiuterinės regos galimybės vis didėja, ją naudojančios sistemos tobulėja ir yra vis plačiau taikomos kasdieniame gyvenime. Ir vis dar egzistuoja daug problemų, prie kurių sprendimo galėtų prisidėti kompiuterinė rega. Įvairi programinė įranga gana lengvai atpažįsta spausdintas raides, skaičius ar kitus simbolius. Tą padaryti sąlyginai lengviau lyginant su ranka parašytais simboliais.

Lietuviškas rankraštis pasirinktas kaip bakalaurnio darbo temos objektas dėl to, jog jo atpažinimas ir taikymas gali būti panaudotas praktiškai Lietuvoje. Viena iš taikymo galimybių – palengvinti mokymosi procesą mokymo įstaigose sunkiau ar lėčiau besimokantiems mokiniams. Taip pat ir studentams, kuriems paskaitų metu kartais reikia spėti greitai užsirašyti didelius informacijos kiekius. Toks pritaikymo būdas leistų labiau susikonscentruoti į ugdytojo pateikiamą informaciją siekiant ją suprasti. Tuo pačiu metu būtų galima išsaugoti ranka užrašytą tekstą skaitmeniniu formatu ir jį vėliau panaudoti mokymosi tikslais.

Šio darbo tikslas:

- Sukurti kompiuterinės programos, kuri atpažįsta lietuvišką rankraštį, prototipą. Programa turi gebėti iš nuotraukų su ant lentos užrašytu tekstu atpažinti ir vartotojui pateikti skaitmeninį teksto variantą.

Uždaviniai:

- Susipažinti su kompiuterinės regos įrankiais naudojamais vaizdo normalizavimui, triukšmo pašalinimui.
- Susipažinti su elementariais mašininio mokymosi įrankiais ir algoritmais.
- Sukurti mokymosi duomenų aibę ir su ja apmokyti teksto atpažinimo algoritmą.

- Sukurti algoritmą, kuris nuskaito ir kuo tiksliau atpažįsta lietuvišką tekstą iš nuotraukos.

Teorinė šio darbo reikšmė yra sužinoti, kokios vaizdo apdorojimo procedūros ir įrankiai yra būtini norint normalizuoti atvaizdą, iš jo pašalinti aplinkos triukšmus. Įgyti bazines žinias apie duomenų rinkimą, paruošimą mašininio mokymosi algoritmams ir įgyti bazines žinias apie pačius algoritmus. Taip pat praplėsti supratimą apie panašių problemų sprendimus naudojantis kompiuterine rega.

Sukurti lietuvišką rankraštį atpažįstančios programos prototipą yra praktinė šio darbo reikšmė. Ši programa suteiktų galimybę greitai ir sklandžiai išsaugoti ranka parašytą tekstą skaitmeniniu formatu. Tai leistų ją panaudoti švietimo srityje moksleiviams, studentams. Taip pat darbas yra orientuotas į lietuvių kalbos rašmenis, todėl jis gali būti naudingas ir sprendžiant teksto vertimo į kitas kalbas problemas.

# 1 Kompiuterinės regos biblioteka OpenCV

Pagrindinis šio bakalauro darbo įrankis naudotas programos prototipo kūrimui yra OpenCV biblioteka [9]. Ši biblioteka – vienas populiariausių įrankių kompiuterinės regos programoms kurti. Ji yra atvirojo kodo, turi aktyvią vartotojų bendruomenę. Biblioteką sudaro daugiau nei 2500 algoritmų, iš kurių daugiausia skirti vaizdų apdorojimui, taip pat mašiniam mokymui. Nors ir parašyta C++ programavimo kalba, OpenCV turi C, Python, Java ir MATLAB kalbų sąsajas taip dar palengvindama naudojimąsi. Taip pat palaikomos Windows, Linux, Mac OS ir Android operacinės sistemos. Daugiausiai naudojama kuriant realiu laiku veikiančias programas.

Buvo pasirinkta naudoti Python programavimo kalbos sąsają, kadangi ji yra paprasta naudoti ir darbo autorius jau turėjo patirties programuojant šia kalba. OpenCV biblioteka turi labai įvairių funkcijų ir pritaikymo būdų. Kadangi darbe buvo naudota tik nedidelė jų dalis, būtent jos yra apžvelgtos toliau.

- Atvaizdo spalvų erdvės transformacijos. Spalvų erdvė yra metodas, kuriuo galima nustatyti, kurti ir atvaizduoti spalvas. Kiekvieną atvaizdą galima apibūdinti bet kurioje spalvų erdvėje [11]. Priklausomai nuo išsikerto uždavinio, svarbu pasirinkti tinkamą spalvų formatą. Dažniausiai sutinkamas formatas, įprastas kompiuteriuose, televizoriuose, skaitmeninėse vaizdo kamerose yra RGB. Juo nurodoma, koks kiekis raudonos, žalios ir mėlynos spalvos reikalingas, norint atitikti konkrečią spalvą. Tuo tarpu HSV erdvė aprašomos spalvos pagal jose esantį pilkos spalvos kiekį ir pagal jų šviesumo stiprumą. Dar kompiuterinėje regoje dažnai naudojama ir pilkumo tonų (angl. grayscale) erdvė. Joje aprašoma tik juodos spalvos vertė.
- Kita svarbi dažnai naudojama vaizdo apdorojimo operacija yra atvaizdo slenksčio (angl. image threshold) nustatymas. Šio metodo metu kaip įvestis naudojamas paveikslėlis pilkumo tonų spalvų erdvėje. Jame, pagal pasirenkamą arba algoritmo nustatomą spalvos vertę kiekvienas taškas įgauna nulio arba vienetą vertę. Tai reiškia, kad už slenkstinę vertę didesnę vertę turintys taškai įgis juodos spalvos reikšmę, ir atvirkščiai.



Pav. 3: Pilkumo tonų atvaizdas kairėje, slenkstinis – viduryje. Invertuotas slenkstinis – dešinėje [12].

- Vartotojo veiksmų fiksavimas, konkrečiau – paspaustų mygtukų fiksavimas yra dar viena OpenCV bibliotekos funkcija, kuria naudotis labai paprasta.

```

1      # Importuoti OpenCV biblioteką:
2      import cv2
3
4      # Kintamajam key priskiriame pirmo paspausto mygtuko reikšmę
5      key = cv2.waitKey(0)
6      # Jei vartotojas paspaudžia "Enter", išspausdinti:
7      if key == 13:
8          print("Paspaudėte Enter.")

```

1 kodo fragmentas. Mygtukų paspaudimo naudojimo pavyzdys.

- Morfologinės transformacijos yra paprastos operacijos, atliekamos remiantis atvaizdo forma. Dažniausiai šioms operacijoms atlikti naudojami binariniai atvaizdai. Kaip įvesties elementai morfologinėms transformacijoms naudojamas norimas atvaizdas ir papildomas struktūrinis elementas, kuris ir nusprendžia, kaip bus vykdoma operacija. Struktūrinis elementas išties yra kvadratinė matrica, dažniausiai užpildyta vienetais. Pagrindinės operacijos yra ardymas (angl. erosion) ir išplėtimas (angl. dilation).

Ardymo operacija veikia panašiai kaip ir dirvožemio ardymas, tik ji ardo atvaizdo priekinio plano objekto kraštus. Struktūrinis elementas slenkamas per paveikslėlio taškus. Originaliam paveikslėlio taškui (turinčiam vertę 0 arba 1) bus priskiriama vertė 1 tik tuo atveju, jei visi taškai po struktūriniu elementu bus lygūs 1. Kitu atveju paveikslėlio taškui priskiriama 0 vertė (jis suardomas). Taip sumažinamas priekinio vaizdo objekto plotas.



Pav. 4: Kairėje – originalus atvaizdas. Dešinėje – po ardymo operacijos.

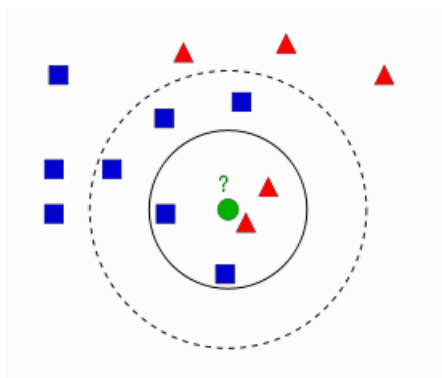
Išplėtimo operacija yra atvirkščia ardymui. Šiuo atveju, jei bent viena taško vertė struktūriniame elemente yra lygi vienam, tuomet ir originalus taškas atvaizde bus lygus vienam. Taigi, šiuo metodu išplečiame priekinio vaizdo objekto plotą. Dažniausiai, kai norima sumažinti triukšmą paveikslėlyje, po ardymo vykdomas išplėtimas. Taip ardymas pašalina baltus triukšmo taškus atvaizde, bet ir sumažina priekinio objekto plotą. Tam, kad objekto plotas vėl padidėtų, atliekamas išplėtimas, bet bet triukšmas nebeatsiranda iš naujo.



Pav. 5: Kairėje – originalus atvaizdas. Dešinėje – po išplėtimo operacijos.

- Mašininio mokymo algoritmai. OpenCV biblioteka turi funkcijų įvairioms mašininio mokymo operacijoms atlikti – statistinius modelius, kNN (angl. K-Nearest Neighbors) algoritmą, SVM (angl. Support Vector Machine) algoritmą, sprendimų medžių, atsitiktinių medžių algoritmus, neuroninius tinklus ir kitus.

Plačiau apžvelgiamas kNN algoritmas. Jis yra vienas paprasčiausių klasifikacijos algoritmų naudojamų prižiūrimam mokymui (angl. supervised learning). Pagrindinė algoritmo idėja yra ypatybių erdvėje surasti artimiausią atitikmenį testiniams duomenims [13]. Pav. 4 dvi skirtingas duomenų klases vaizduoja mėlyni kvadratai ir raudoni trikampiai. Kiekvienas duomuo turi po 2 ypatybes (grafike tai parodo x ir y koordinatės ir taško pozicija). Naujas testinis duomuo yra žalias apskritimas. Tad proceso, vadinamo klasifikacija metu algoritmas turi nuspręsti, kuriai duomenų klasei jį priskirti. Jei žiūrima tik į vieną artimiausiai esantį kaimyną, tai testinis duomuo būtų priskirtas raudonų trikampių klasei (kaip ir jei ieškoma 2 artimiausių kaimynų). Klasifikuojant pagal 3 kaimynus, duomuo vis tiek būtų priskirtas raudonųjų klasei. Naudojant šį algoritmą nereikėtų rinktis lyginio kaimynų skaičiaus, nes tuomet rezultatas gali būti lygus ir reikėtų ieškoti kitų savybių norint teisingai klasifikuoti. Pasirinkus 5 ar daugiau kaimynų, matoma, kad tokiu atveju žaliasis apskritimas būtų priskirtas mėlynųjų klasei. Toks klasifikavimas žiūrint į duomenų išsidėstymą atrodo teisingas.



Pav. 6: kNN algoritmo vizualizacija 2 duomenų klasėse.

Galima tarti, kad kNN algoritmui įtaką daro tik artimiausių kaimynų skaičius. Tačiau, būtina paminėti, kad klasifikuojant kiekvieną naują duomenų aibės narį, esamiems duomenims priskiriami skirtingi svoriai. Jie nustatomi pagal atstumą iki naujojo nario. Remiantis šia savybe, galima tarti, kad klasifikuojant pav. 4 grafike esantį žalią ap-



skritimą pagal 4 artimiausius kaimynus, jis būtų priskirtas raudonųjų trikampių klasei, kadangi šie yra arčiau nei mėlynieji kvadratai.

## 1.1 A1 pavadinimas

Tekstas ...

## 1.2 A2 pavadinimas

Tekstas su formule  $y = x^2$ ...

# 2 Problemos sprendimas

Norint sukurti programos, gebančios atpažinti lietuvišką rankraštį, prototipą ir jį pritaikyti, buvo pasirinkta Python programavimo kalba. Naudojantis įvairiais kalbos moduliais bei OpenCV kompiuterinės regos biblioteka atlikti išsikelti uždaviniai ir įgyvendintas pagrindinis darbo tikslas – sukurtas ir praktiškai išbandytas prototipas. Šiame skyriuje aprašomas uždavinių sprendimas, jų rezultatai, kilusios problemos, pastabos ir pasiūlymai.

## 2.1 Pradinis vaizdo apdorojimas

Norint išbandyti tinkamą programos prototipo veikimą, iš pradžių būtina turėti kokybiškos vaizdinės medžiagos. Todėl šiame darbe naudota vaizdinė medžiaga buvo fiksuojama esant geroms, pastovioms apšvietimo sąlygoms. Tokiu būdu norėta išvengti papildomo triukšmo paveikslėliuose. Taip pat buvo stengiamasi išrinkti pakankamai kontrastingus priekinio plano objektus ir foną atvaizduose.

Darbe naudojami atvaizdai pirmiausiai įkeliami į programą naudojant OpenCV bibliotekos funkcijas. Taip pat sumažinama jų skiriamoji geba.

```
1      # Importuoti OpenCV biblioteką:
2      import cv2
3      # Įkelti atvaizdą į programą:
4      img = cv2.imread('gb3.png')
5      # Sumažinti skiriamąją gebą:
6      newx,newy = img.shape[1]/3.5,img.shape[0]/3.5
7      newimage = cv2.resize(img,(int(newx), int(newy)))
```

2 kodo fragmentas. Atvaizdo įkėlimas į programą.

## 2.2 B2 pavadinimas

Poskyris B2 turi du skirsnius

### 2.2.1 B21 pavadinimas

Tekstas....

### 2.2.2 B22 pavadinimas

Tekstas....

## 2.3 B3 pavadinimas

Tekstas su nauja formule  $y = x^3 \dots$

## Literatūros sąrašas

- [1] G. Bradski ir A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*, O'Reilly Media, Inc., 2008, p. 1–8.
- [2] N. Linder, ir kt., *A malaria diagnostic tool based on computer vision screening and visualization of Plasmodium falciparum candidate areas in digitized blood smears*, PLoS One 9.8 (2014): e104855.
- [3] E. Christos-Nikolaos Anagnostopoulos ir kt., *License plate recognition from still images and video sequences: A survey* IEEE Transactions on intelligent transportation systems 9.3 (2008), p. 377-391.
- [4] J. Pradeep, E. Srinivasan ir S. Himavathi, *Diagonal based feature extraction for handwritten character recognition system using neural network*, Electronics Computer Technology (ICECT), 2011 3rd International Conference IEEE, 2011.
- [5] C. Chen ir kt., *A cost-effective people-counter for a crowd of moving people based on two-stage segmentation*, Journal of Information Hiding and Multimedia Signal Processing 3.1 (2012): 12-25.
- [6] Y. LeCun ir kt., *Comparison of learning algorithms for handwritten digit recognition*, Tarptautinė neuroninių tinklų konferencija, 1995. <http://yann.lecun.com/exdb/publis/pdf/lecun-95b.pdf>
- [7] D. Kostadinov, *Privacy Implications of Automatic License Plate Recognition Technology*, <http://resources.infosecinstitute.com/privacy-implications-automatic-license-plate-recognition-technology/>.
- [8] Shujing Lu ir kt., *Cost-sensitive neural network classifiers for postcode recognition*, International Journal of Pattern Recognition and Artificial Intelligence, 2012.
- [9] OpenCV kompiuterinės regos bibliotekos žiniatinklio puslapis, <http://opencv.org/about.html>

- [10] OpenCV spalvų erdvės transformacijos, [http://docs.opencv.org/3.2.0/df/d9d/tutorial\\_py\\_colorspaces.html](http://docs.opencv.org/3.2.0/df/d9d/tutorial_py_colorspaces.html)
- [11] A. Ford, Alan Roberts, *Colour Space Conversions*, 1998 <http://sites.biology.duke.edu/johnsenlab/pdfs/tech/colorconversion.pdf>
- [12] Atvaizdo slenkstinės vertės nustatymas, [http://docs.opencv.org/trunk/d7/d4d/tutorial\\_py\\_thresholding.html](http://docs.opencv.org/trunk/d7/d4d/tutorial_py_thresholding.html)
- [13] kNN algoritmo paaiškinimas, [http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_ml/py\\_knn/py\\_knn\\_understanding/py\\_knn\\_understanding.html#knn-understanding](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_ml/py_knn/py_knn_understanding/py_knn_understanding.html#knn-understanding)

## Santrauka

Trumpa darbo santrauka...

## Summary

Short english summary...