Äärmiselt pikselleeritud teksti lugemine optilise märgituvastuse abil

Bakalaureusetöö laiendatud kokkuvõte

Lasse Thor Lepik, juhendaja Andres Käver

1. Teema kirjeldus, valiku põhjendus

Tausta kirjeldus

Optilise märgituvastuse tehnoloogiat kasutatakse, et automatiseeritud viisil lugeda fotodelt või skaneeritud dokumentidelt teksti, tuvastada numbrimärke või digitaalselt talletada käsitsi kirjutatud infot. Optiline märgituvastus on samuti kasutusel pildipõhiste CAPTCHA-süsteemide petmiseks ning peale moonutatud teksti võimaldab ka lugeda pikselleeritud teksti. Sattusin töö teema peale lugedes Factorio arendusblogi FFF-380, kus arendaja oli avaldamata info pikselleerides peitnud ja lugejad püüdsid seda lahti murda.

Teema olulisus/aktuaalsus

Madala eraldusvõimega pildid ei võimalda neilt selgelt teksti lugeda. Selguskadu põhjuseks võib olla tahtlik andmete pikselleerimine või lihtsalt eraldusvõime kadu liigse töötlemise tõttu, mille üks põhjus võib olla faili korduv ringlemine sotsiaalmeedias. Teksti tuvastamine on eriti oluline valdkondades nagu turvalisus ja luure. Mitte ainult ei võimalda OCR siinkohal piltidelt info lugemist automatiseerida, vaid hea süsteemi puhul tuvastada inimese jaoks loetamatut teksti [1].

Asjakohased mõisted

- **Optiline märgituvastus (OCR)**: Tehnoloogia, mis võimaldab pildil kujutatud teksti digitaalseks sõneks muuta**.**
- **Närvivõrgud**: Masinõppe mudelid, mis on inspireeritud inimese ajust ja suudavad tuvastada keerukaid mustreid.
- **Pikselleerimine**: Pildi või selle sektsiooni eraldusvõime vähendamine, kasutades näiteks algoritmi *„block averaging”*.
- **Block averaging**: Pilditöötlusmeetod, mis jagab pildi väikesteks plokkideks ja asendab iga ploki pikslid selle ploki keskmise värviga.

Lahtised küsimused/probleemid

- Kuidas tuvastada teksti, mis on pikselleeritud 2-piksli kõrguseks ja on inimsilmale loetamatu? Kas on võimalik vähesel määral teksti tuvastada, kui kõrgus on 1 piksel?
- Milliseid masinõppe meetodeid ja mudelite arhitektuure on tulus rakendada nii madala eraldusvõimega piltide puhul ja kuidas mõjutab monokroomsete piltide kasutamine mudeli jõudlust võrreldes värviliste piltidega?
- Kui hästi suudab üks mudel tuvastada erinevate fontide, renderdus-mootorite ja pikselleerimis-algoritmide abil loodud teksti (generaliseerimine)? Kas on kasulik teha

spetsialiseerunud mudelid? Kas spetsialiseerunud mudeli automaatseks valimiseks peaks looma otsuseid tegeva kontroller-mudeli?

- Kui edukalt on võimalik lugeda täiesti suvalist andmejada, mille puhul ei saa rakendada statistikal põhinevat meetodit „*Hidden Markov model*“?
- Teadustöid on tehtud, aga kust saab keskmine kasutaja kätte hästi töötava tarkvara, mis ta reaalse probleemi lahendaks?

2. Kirjanduse ülevaade

On kirjutatud mitmeid teadustöid, mis püüavad üldises võtmes lahendada sama probleemi. Töödes kasutati erinevaid lähenemisviise ning uuriti nende efektiivsust.

Närvivõrgud:

Tipptulemusi saavutavad närvivõrgud, mis kasutavad implementatsioonis *bidirectional LSTM (Long Short-Term Memory)* ja *CTC (Connectionist Temporal Classifier)* komponente. [1, 2]

Hidden Markov model:

HMM aitab närvivõrgul teha täpseid ennustusi keelelise statistika põhjal [3]. Näiteks kui on tegu udustatud sõnega „MOON“, siis närvivõrk võib pakkuda pikslite põhjal „M00N“ ja HMM aitab närvivõrgul eelistada sarnast, aga statistiliselt tõenäolisemat varianti „MOON“. Kui tegeleme suvalise andmejadaga, siis pole sellest lähenemisest kasu.

Toore jõuga proovimine ja võrdlemine:

Tööriistad nagu Depix ja Unredacter püüavad genereerida sarnaseid vasteid pikselleeritud pildile. Need meetodid on aeglased ja pole ideaalsed. Depix vajab *De Bruijn* jada ning Unredacter’il on hulk probleeme tähtede paiknemise ja segmenteerimisega. [4, 5]

Värvide roll masinõppes:

Monokroomsete piltide kasutamine muudab teksti ääred mudeli jaoks selgemaks, aitab paremini generaliseerida, muudab mudeli müra vastu robustsemaks ja võtab vähem ressursse, kuid paljude sarnaste klasside ja eriliste kasutusalade puhul võib värviinfo mudeli täpsust parandada. [6]

3. Probleemi püstitus

Eeltoodud tööd näitavad, et madala eraldusvõimega tekstide tuvastamise valdkonnas on tehtud edusamme, kuid pole ideaalset, lihtsat, kättesaadavat ja paindlikku tarkvara, mis loeks pildilt usaldusväärselt seosetut teksti, mis on 2-pikslit kõrge.

Töö eesmärk

Arendada kättesaadav tarkvara, mis kasutab närvivõrgu mudelit, et tuvastada inimsilmale loetamatut pikselleeritud teksti. Eesmärgiks on spetsiifilistel tingimustel lugeda 2-piksli kõrgust teksti ja väga piiratud stsenaariumites äkki isegi ühe piksli kõrgust teksti. Töö käigus analüüsin mudelite efektiivsust ja piiranguid.

Töö panus/uudsus

- Valmib avatud lähtekoodiga lihtsalt kasutatav ja kiire tarkvara, mis ei toetu vaid toorele jõule ega HMM tehnikale.
- Mitmed spetsiaalsete tingimuste jaoks loodud mudelid, näiteks erinevate teksti-suuruste, fontide ja märgijadade (tähed, numbrid) jaoks.
- Potentsiaalne kasu kontroller-mudelist, mis aitab automaatselt valida parima OCR mudeli kasutaja sisendpildi lugemiseks.
- Eksperimenteerimine madala eraldusvõimega teksti värviinfo rakendamises.
- Spetsiaalne närvivõrgu arhitektuur äärmuslike tingimuste jaoks ja hüperparameetrite tuunimine.

4. Metoodika valik ja põhjendus

Uurimisstrateegia:

Käesolev töö kasutab eksperimentaalset uurimisstrateegiat, mis keskendub uute masinõppe mudelite väljatöötamisele ja hindamisele äärmiselt pikselleeritud teksti tuvastamisel. Eksperimentaalse lähenemise valik võimaldab katsetada erinevaid mudeliarhitektuure ja meetodeid, et leida optimaalseim lahendus antud probleemile. Sünteesitakse suur hulk testandmeid, mis hõlmavad erinevaid teksti suurusi, fonte ja pikselleerimisalgoritme. Sünteetiliste andmete peal tehtud testide põhjal tehakse tarkvarale täiustusi.

Meetodite valik:

- **Närvivõrgu loomine ja treenimine**: Kasutatakse spetsiaalselt kohandatud konvolutsiooniliste ja LSTM-kihtide kombinatsiooni koos teiste lisakihtidega. Võrdluseks võetakse paar mudelit, mida on kasutatud piksellatsiooniga seotud teadustöödes või CAPTCHA lahendamiseks. Närvivõrgud on paindlikud, võimelised leidma andmetes vaevumärgatavaid mustreid ning skaleeruvad võrreldes teiste efektiivsete lahendustega paremini.

Ülevaade teistest meetoditest:

- **Suvaline lahendus + HMM**: Kui tekstis puuduvad mustrid, on *Hidden Markov model* kasutu. Mustrite esinemise puhul on kasulik lisada see närvivõrgule.
- **Toores jõud**: Võrreldakse pikselleeritud pilte sildistatud piltidega, mis on renderdatud varem ette või reaalajas. Võrreldakse pikslite heledust, et leida milline silt vastab sisendpildile. Paralleeli saab tuua räsimisega: loome suvalisi andmeid kuni räsi tuleb sama. Võtab massiivselt aega, arvutusjõudu ja skaleerub väga halvasti.
- **Toores jõud + segmenteerimine**: Proovitakse jagada pilt tähtedeks ja teha igale tähele eraldi otsing. Madala eraldusvõime puhul on raske tähti eraldada, erinevate tähtede pikslid on kokku sulanud. Kui on olemas teksti *De Bruijn Sequence,* võib see probleemi veidi lihtsustada.

- **Super-resolutsiooni meetodid**: Võiksid potentsiaalselt parandada pildi kvaliteeti enne OCR-i rakendamist, kuid nende efektiivsus 2-piksli kõrgusel pildil on pea olematu.
- **Traditsioonilised pilditöötlusmeetodid**: Ei anna inimesele lisainfot, mille põhjal ta saaks piksleid tekstiks tõlkida. Inimsilma kasutades on väga raske pikselleeritud teksti lugeda.

5. Töö ülesehitus

Töö koosneb järgmistest etappidest:

1. **Andmete ettevalmistamine**: Luuakse programm, mis sünteesib suure hulga andmeid, mida kasutatakse treenimiseks ja testimiseks. Andmeteks on paarid pikselleeritud piltidest ja piltidel kujutatud tekstidest.
2. **Närvivõrgu arhitektuuri väljatöötamine**: Luuakse eksperimentaalne arhitektuur.
3. **Mudelite treenimine**: Sünteesitud andmete põhjal treenitakse mudelid.
4. **Tulemuste analüüs**: Hinnatakse mudelite täpsust ja valitakse parimad arhitektuurid ja meetodid.
5. **Tsükliline arendus**: Tulemuste põhjal kohandatakse meetodeid ja koodi. Etapid käiakse uuesti läbi.
6. **Viimistlemine**: Kui programm töötab edukalt, siis parandatakse jõudlust ning tehakse tarkvara kasutaja jaoks lihtsamaks.

6. Tulemuste valideerimine

Mudeli efektiivsust hinnatakse järgmiselt:

- **Manuaalne valideerimine testandmestikul**: Kasutatakse sisendeid, mida mudel pole treeningu ajal näinud, et veenduda tarkvara võimes lahendada probleemi.
- **Automatiseeritud jõudlustest**: Mudel töötab läbi tuhandeid pilte ja programm koostab statistilise ülevaate, kus on näiteks antud protsentides keskmine *Sequence Error Rate* (kui sageli ei loe mudel sõne perfektselt) ja *Levenshtein distance* (kui erinev on loetud sõne).

Kontseptsiooni tõestamiseks loodud prototüüp-mudeli ennustuste jaoks vaata lisasid 1 ja 2.
Sünteetiliste andmete peal saavutas mudel mõne tunniga protsessoril treenides SER 1.5%.

7. Viited

[1] V. Garske and A. Noack, "Recovering information from pixelized credentials," in GI SICHERHEIT 2022, Karlsruhe, Germany, Apr. 2022, pp. 129-141, doi: 10.18420/sicherheit2022_08.

[2] J. Gilbey and C.-B. Schönlieb, "An end-to-end Optical Character Recognition approach for ultra-low-resolution printed text images," submitted for publication, May 2021.

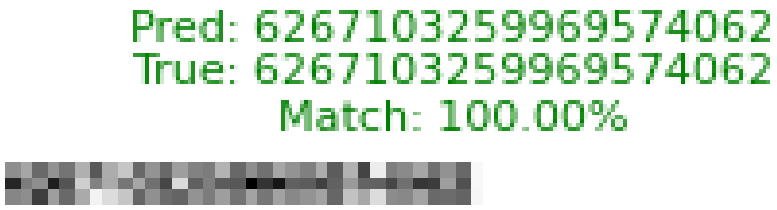
[3] S. Hill, Z. Zhou, L. Saul, and H. Shacham, "On the (in)effectiveness of mosaicing and blurring as tools for document redaction," in *Proceedings on Privacy Enhancing Technologies*, 2016, doi: 10.1515/popets-2016-0047.

[4] S. Mellema, "Recovering passwords from pixelized screenshots," Technical blog, Dec. 2020. [Online]. Available: https://www.spipm.nl/2030.html

[5] D. Petro, "Never, Ever, Ever Use Pixelation for Redacting Text," Tech blog, Feb. 15, 2022. [Online]. Available: https://bishopfox.com/blog/unredacter-tool-never-pixelation'

[6] V. Buhrmester, D. Münch, D. Bulatov, and M. Arens, "Evaluating the Impact of Color Information in Deep Neural Networks," in *Pattern Recognition and Image Analysis. IbPRIA 2019*, A. Morales, J. Fierrez, J. Sánchez, and B. Ribeiro, Eds., Lecture Notes in Computer Science, vol. 11867, Cham: Springer, 2019, pp. 27, doi: 10.1007/978-3-030-31332-6_27.

8. Lisad



Lisa 1. Eksperimentaalse mudeli ennustus pikale numbrijadale pildi.

Lisa 2. Eksperimentaalse mudeli ennustus kahe piksli kõrguse pildi numbrijadale.