VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Kursinis darbas

Gestų kalbos atpažinimas naudojant internetinę kamerą

(Sign language recognition using web camera)

Atliko: 3 kurso I grupės studentas
Pranciškus Ambrazas (parašas)

Darbo vadovas:
asist. Linas Petkevičius (parašas)

Vilnius, 2017

Turinys

Įvadas

Daugiau nei 360 milijonų žmonių pasaulyje kenčia dėl klausos ir kalbos įvairių problemų, o daugiau nei 32 milijonai jų yra vaikai ir šis skaičius vis auga [4]. Gestų kalba yra pagrindinis šių žmonių bendravimo įrankis. Tačiau reiktų pastebėti ir tai, jog dalis jų moka dalinai skaityti iš lūpų. Norint žmogui be šių ydų bendrauti su gestakalbiu (*gestų kalba kalbantis žmogus*) reikia vertėjo, kuris išverstų gestų kalbą į įprastinę ir atvirkščiai.

Kiekviena pasaulyje esanti kalba turi ir savo gestų kalbą. Tai reiškia, skiriasi tiek gestų kalbos gramatika, tiek netgi patys gestai. Pasaulyje randama net dialektų pagal regionus, ne tik pagal šalis. Amerikiečių anglų gestų kalba (*toliau - ASL*) šnekančių žmonių pasaulyje priskaičiuojam nuo 500 tūkstančių iki netgi 2 milijonų vien tik Jungtinėse amerikos valstijose. Remiantis Census Bereau surinktais duomenimis, kuris domisi kalbų mažumomis, ASL yra pirmoji mažumos kalba, po „didžioje ketverto“, kurį sudaro ispanų, italų, vokiečių ir prancūzų kalbos [3]. Tad netgi bendraujant dviem žmonėms, mokantiems gestų kalbą neretai iškyla vertimo problema, todėl tenka ieškoti gestų vertimų. Paiešką šiuo metu galima atlikti atsižvelgiant į delno padėtį, vienos ar abiejų rankų judesį, jų padėtį ir gestą atliekančių rankų skaičių. Tuomet pagal gesto išvaizdos nuotraukas ar kartais net vaizdo įrašus, gestakalbiai gali išsiversti gestus. Tam yra skirtos tiek internetinės svetainės - žodynai, tiek įvairūs rašytiniai žodynai.

Šio tyrimo tikslas - ištirti ir išanalizuoti galimybes internetinės kameros pagalba versti gestų kalbą. Taip padedant ne tik gestakalbiams tarpusavyje, bet ir žmonėms, nesuprantantiems gestų kalbos bendrauti su gestakalbiais tam pasitelkiant technologijas, taip suteikiant šiems žmonėms pilnavertį gyvenimą bendraujant su kitais. Šiuo tyrimu siekiama apžvelgti ir įvertinti ar naudojantis įprasta internetine kamera įmanoma paversti gestų kalbą rašytiniu tekstu ar net garsine kalba ir lygiai taip pat versti rašytinę ar garsinę kalbą į gestų kalbą. Taip pat siekiama, kad vėliau būtų sukurtas visiems gestakalbiams prieinamas produktas ar programinė įranga, kurią kiekvienas įsidiegęs į savo įrenginį - kompiuterį, mobilųjį telefoną ar planšetinį kompiuterį galėtų naudotis šiuo vertėju. Vėliau tai galėtų tapti ir mokomąja gestų kalbos priemone.

Naudojamos priemonės:

1. Programavimo kalba Python;
2. Vaizdų apdorojimo įrankis OpenCV;
3. Matematinių skaičiavimų biblioteka NumPy;
4. Įrankių moksliniams tyrimams biblioteka SciPy;
5. Savaime apsimokančių sistemų biblioteka scikit-learn;

Tyrimo eiga

Šiame darbe bus nagrinėjamos amerikiečių anglų ir lietuvių gestų kalbos.

1. Susipažinimas su gestų kalba ir jos problematika

2. Sistemos apmokymas atpažinti gestus
3. Gestų kalbos statinių ženklų atpažinimas
4. Gestų kalbos dinaminių ženklų atpažinimas

1. Teorija

Šiame skyriuje bus aprašoma teorija apie apsimokančias sistemas ir neuroninius tinklus

1.1. Savaime apsimokančios sistemos

Apsimokančios sistemos (*angl. machine learning*) - tai mokslas apie tai, kaip kompiuterius užprogramuoti taip, jog jie patys darytų sprendimus be žmogaus įsikišimo neužprogramuojant kiekvienos galimos situacijos. Kitais žodžiais tariant, leisti kompiuteriui pačiam nuspręsti kaip elgtis esant tam tikroms aplinkybėms. Savaime apsimokančios sistemos yra didelis žingsnis į priekį norint sukurti dirbtinį intelektą.

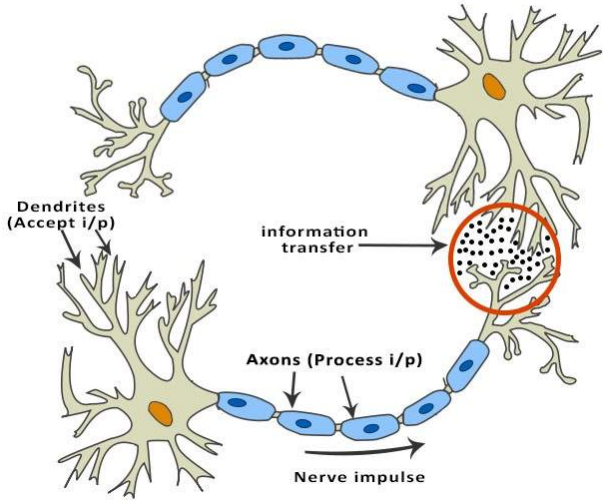
Savaime apsimokančių sistemų ir jų algoritmų sukūrimo dėka gatvėmis pradėjo važinėti patys save vairuojantys automobiliai (*angl. self-driving cars*) arba dar kitaip vadinamos autonominės transporto priemonės. Įvairūs paieškos varikliai tokie kaip „Google“ ar „Yahoo“ taikydami šiuos alogirtmus naudotojams rodo kiekvienam asmeniškai sugeneruotą turinį. Taip pat reikėtų paminėti ir kalbos atpažinimo sistemas tokias kaip „Siri“ ar „Google Assistant“, kurios iš joms duotų komandų atlieka tam tikrus veiksmus.

1.2. Neuroniniai tinklai

Neuronas (*arba - nervinė ląstelė*) - pagrindinės nervų sistemos ląstelės, sukuriančios ir/arba perduodančios elektrocheminius impulsus.

Žmogaus smegenys yra sudėtingas, nelinijinis ir paralelinis kompiuteris[2], kurias sudaro neuronai. Vienas neuronas vienu metu jungiasi su daugybe kitų neuronų per dendritus, ant kurių yra daug sinapsių, per kurias ateina informacija iš kitų neuronų. Dendritus paprasčiau galima pavadinti kaip informacijos priimėjais. Todėl vienas neuronas gali sudaryti iki 100000 sunapsių. Kiekviena sinapsė gali būti jaudinanti arba slopinanti. Visas šis mechanizmas dar nadinamas **neuroniniais tinklais** (*angl. Neural network*).

Vienas neuronas priima informaciją per dendritus, tuomet pats neuronas nusprendžia ar bus siunčiama informacija į kitus neuronus ir kokia ji bus siunčiama (*žr. 1 pav.*)



1 pav. Neuroninio tinklo struktūra[1]

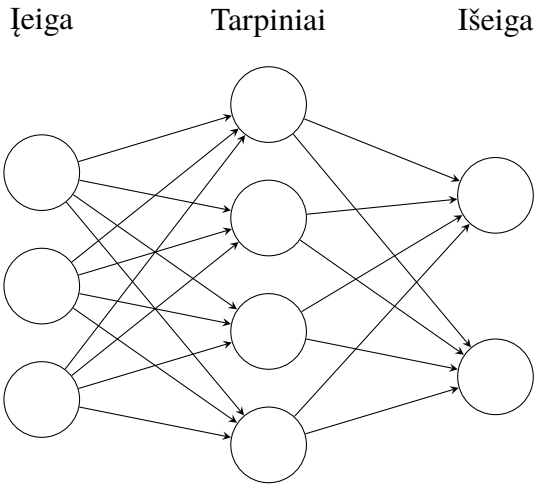
Remiantis šiais principais buvo sukurti dirbtiniai neuroniniai tinklai.

1.2.1. Dirbtiniai neuroniniai tinklai

Dirbtinis neuroninis tinklas (*angl. Artificial neural network*) - struktūra, skirta apdoroti dideliam kiekiui informacijos, sukurta rementis žmogaus nervų sistemos veikimo principu. Kitais žodžiais tariant, skaitmenizuota žmogaus smegenų veikla.

Neuronai dirbtiniame neuroniniame tinkle yra sujungti jungtimis. Taip jie tarpusavyje komunikuoja perduodami vienas kitam informaciją. Kiekvienas neuronas gali priimti atėjusią informaciją, ją apdoroti ir perduoti kitam neuronui. Kiekviena jungtis turi savo svorį, pagal kurį pasirenkama, į kurį neuroną turi būti perduodama informacija. Šių jungčių svorius galime įvertinti naudodamiesi apsimokančių sistemų pagalba.

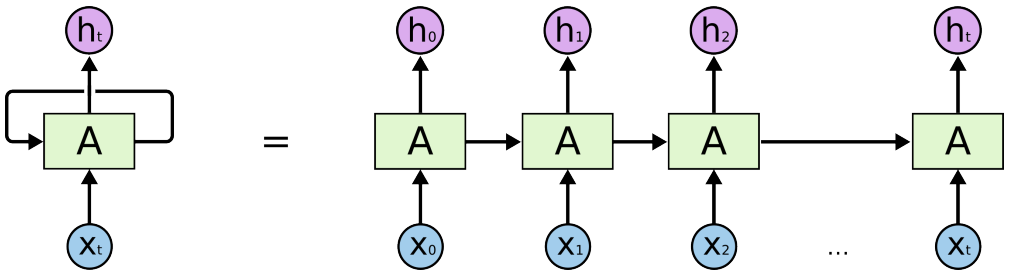
?? pav. pateikiama paprasta schema, kaip veikia dirbtiniai neuroniniai tinklai:



[1] `https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.htm/`

1.2.2. Rekurentiniai neuroniniai tinklai

Rekurentiniai neuroniniai tinklai (*angl. Recurrent neural network*) - tai dirbtinis neuroninis tinklas, kuris saugo informaciją apie praeituose žingsniuose (neuronuose) atliktus veiksmus ar skaičiavimus.



2 pav. Rekurentinių neuroninių tinklų schema[2]

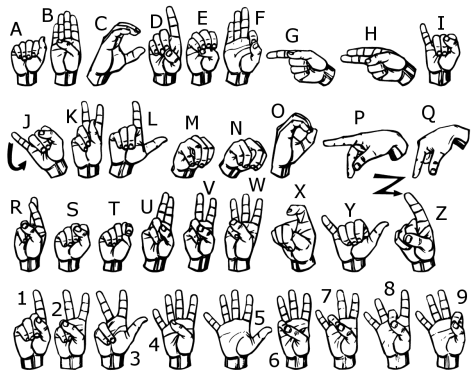
- x_t - įeiga momentu t
- h_t - išeiga momentu t
- A_t - būsena momentu t

[2] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

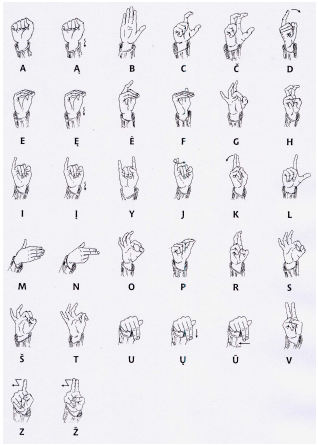
2. Gestų kalba

2.1. Gestų kalbos skirstymas

Gestų kalba susideda iš dviejų dalių - statinių ir dinaminių ženklų. Gestų kalboje kiekviena kalba turi savo abėcėlę. Statiniais ženklais atvaizduojama didžioji abėcėlių raidžių dalis. O dinaminiais - žodžiai ir kai kurios gestų abėcėlių raidės. *Pavyzdžiui*, amerikiečių gestų kalbos abėcėlėje J ir Z raidės atvaizduojamos dinaminiais judesiais (*žr. 3 pav.*), o lietuvių - Ą, D, Į, K ir kt. bei jau minėtosios J ir Z raidės (*žr. 4 pav.*)



3 pav. Amerikiečių gestų kalbos abėcėlė[3]



4 pav. Lietuvių gestų kalbos abėcėlė[4]

2.2. Problematika

Norint atpažinti gestus, paversti juos į raides, žodžius ar sakinius susiduriama su problemomis, kurios susijusios tiek su statinių, tiek su dinaminių gestų atpažinimu.

2.2.1. Statinių ženklų problematika

Pagrindinės problemos iškylančios atpažįstant statinius gestų kalbos ženklus yra:

1. Kiekvienos kalbos abėcėlę sudaro skirtingas raidžių (statinių ženklų) skaičius. *Pavyzdžiui*, lietuvių kalbos abėcėlę sudaro 32 ženklai, o amerikietišką - 26;
2. Gestų panašumai. *Pavyzdžiui*, raidės A, E, N, S, T yra atvaizduojamos sugniaužtus kumštį, o net trijose iš jų (A, E ir S) skiriasi tik nykščio padėtis;
3. Kampas, kuris susidaro atpažįstant gestą. *Pavyzdžiui*, kai A raidė rodoma ne iš priekio, o iš šono;
4. Apšvietimas. *Pavyzdžiui*, gestų atpažinimas esant prieblandai ir dienos šviesai.

[4] http://lifeprint.com/asl101/topics/wallpaper1.htm
[4] http://gestai.ndt.lt/pirstu-abecele

2.2.2. Dinaminių ženklų problematika

Pagrindinės problemos iškylančios atpažįstant dinaminius gestų kalbos ženklys yra:

1. Nauji gestų kalbos žodžiai. *Pavyzdžiui*, kiekvienas uraganas turi savo pavadinimą, todėl tai gali reikšti naujo gesto atsiradimą;
2. Gesto kelios reikšmės. *Pavyzdžiui*, vienas gestas gali turėti kelias reikšmes, kaip kad lietuvių kalboje vienas žodis „kasa“ gali turėti net tris skirtingas reikšmes;
3. Kampas, kuris susidaro atpažįstant gestą. *Pavyzdžiui*, kai A raidė rodoma ne iš priekio, o iš šono;
4. Žodžių apjungimas į vieną sakinį. *Pavyzdžiui*, keli gestai einantys vienas po kito gali reikšti vieną žodį, tačiau tuo pačiu būti panašūs į vieną gestą, kuris jau reikš tik vieną žodį.

3. Sistemos apmokymas

Norint, jog sistema atpažintų gestus, svarbiausia ją apmokyti ką reiškia tam tikri gestai. Tam galime išnaudoti kadrus (*angl. frame*) ir savaime apsimokančių sistemų galimybes. Tad reiktų imti vieną kadrą ir jį paversti į kompiuteriui suprantamą kalbą.

Imkime, jog nuotrauka yra sudaryta iš *n* * *m* taškų (*angl. pixels*). Reikia, jog sistema atskirtų, kurioje kadro vietoje yra rodomo gesto dalis, kurioje tik pašalinis fonas.

Apmokymui toliau aptarsime Sobel branduolį, savaime apsimokančių sistemų metodus ir duomenų surinkimą

3.1. Kadro apdorojimas naudojant Sobel branduolį

Sobel branduolys (*angl. Sobel operator*) - vaizdų apdorojimo algoritmas, skirtas skirtas paversti kadrą į kontūrų žemėlapį.

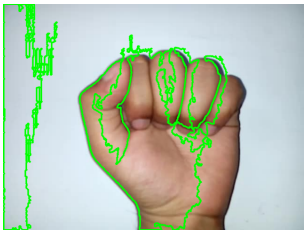
Šis branduolys naudojasi šiomis funkcijomis, kad konvertuotų vaizdą į kontūrus:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \tag{1}$$

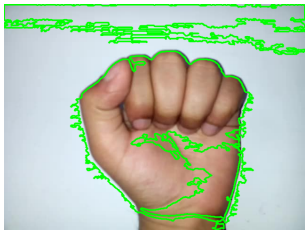
$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \tag{2}$$

$$G = \sqrt{G_x^2 + G_y^2} \tag{3}$$

5 pav. Orginalus paveikslėlis



6 pav. Pritaikyta G_x



7 pav. Pritaikyta G_y

8 pav. Pritaikyta G

9 pav. Be fono

10 pav. Dviejų spalvų

Prieš kadrui taikant Sobel funkcijas, reikia jį paruošti. Kadangi Sobel funkcijos ima matricas

sudarytas iš 3 x 3 skaičių, tai reikia sušvelninti šalia esančius skaičius. Tą padaryti lengviausia pasinaudojant uždedant, pavyzdžiui, 5% miglą (*angl. blur*) kadrui. Uždėti miglą reikia tam, jog sumažintume triukšmą (*angl. noise*) kontūrams.

Toliau, atiduodant kadrą Sobel funkcijoms, kadras yra konvertuojamas į skaičių masyvą, kur kiekvienas taškas turi savo skaičių - spalvos kodą. Tuomet norint rasti visus kontūrus esančius kadre, taikome šiuos veiksmus:

1. Einame per visus taškus esančius kadre ir taikome 1 formulę, kur A - kiekvienas 3 x 3 kadro taško spalvos mastyvas. Gauname tokį vaizdą - žr. 6 pav.;
2. Einame per visus taškus esančius kadre ir taikome 2 formulę, kur A - kiekvienas 3 x 3 kadro taško spalvos mastyvas. Gauname tokį vaizdą - žr. 7 pav.;
3. Apskaičiuojame dabartinio taško tikrąją reiškę taikydami 3 formulę. Gauname tokį vaizdą - žr. 8 pav.

Po šių žinsnių turime matricą taškų, kuriose keičiasi spalva. Dabar reikia išimti tuos taškus, kurie yra, pavyzdžiui, ta pati balta spalva, tik kitokio atspalvio. Tam, kad tai padarytume, imame ir susižinome matricos vidurkį ir atmetame visus taškus, kurie yra mažesni už vidurkį. Kitaip tariant, išimame mažo skirtumo taškus. Taip gauname, jog lieka tik tie taškai, kurie jau turėtų priklausyti gesto kontūrui.

Ieškant kontūro taip pat atmetame ir tuos kontūrus, kurie, pavyzdžiui, neužima daugiau nei 5% viso ploto ir nubrėžiame kontūrą.

Kitas žingsnis paversti kadrą į kompiuteriui suprantamą ir kuo paprastesnę kalbą. Turėdami kontūrus, galime kadrą paversti į dvispalvį kadrą, kur viskas, kas yra kontūre bus baltos spalvos, o viskas kas už kontūro - juodos. Tam reiktų pasidaryti kaukę (*angl. mask*), kurioje viskas, kas už kontūro ribų bus juodos spalvos, tai kas mūsų atveju yra fonas (*žr. 9 pav.*), o paskui ištrinti tai, kas yra kontūre ir padaryti baltos spalvos, tai kas mūsų atveju bus ženklas (*žr. 10 pav.*).

Paskutinis žingsnis, prieš apmokant modelį, reikia kadrą paversti duomenimis, iš kurių modelis galėtų mokytis. Šioje vietoje buvo pasirinktas plačiausiai naudojamas metodas - kadrą paversti į skaičių matricą, kur balta spalva atitinka 255, o juoda - 0. Ši konvertacija pasirinkta pagal spalvų kodus. Jau pavertus kadrą į skaičių matricą, suplokštiname ją, kad gautume vienmatę matricą. Ir galiausiai į vieną bendrą duomenų rinkmeną (*angl. file*) surašome duomenis tokia seka - pirmas eilutės langelis - raidė, visi likę šios eilutės langeliai užpildomi vienmatės kadro matricos duomenis.

3.2. Apsimokančių sistemų apmokymas iš kadro

Turėdami rinkmeną, kurioje yra surašyti visi apmokymams skirtų kadrų duomenys, aptarsime, kaip iš šių duomenų apmokyti sistemą.

3.2.1. Kitos technikos

Šiame poskyryje aptarsime keletą objektams atpažinti populiarių technikų, kuriomis remiantis galime apmokyti sistemą atpažinti gestus.

Aptarsime visas technikas pasinaudodami ASL abėcėlės aibe, kurioje yra 24 statiniai gestai.

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + ... \tag{4}$$

4 formulė apibrėžia, kaip bus apskaičiuojamas parametras *z*, kai norima susidaryti pasirinktos formos plotą, pagal kurį gestas priklausys arba nepriklausys išmoktai abėcėlės raidei.

3.2.1.1. Logistic Regression

Vienas populiariausių šiuo metu naudojamų klasifikavimo algoritmų.

$$0 \leq h_\theta^i(x) \leq 1; i = 1, 2, ..., 24 \tag{5}$$

Remiantis 5 formule, apsibrėžiama, jog tikimybės kiekvienai raidei bus ne mažesnės nei 0 ir ne didesnės nei 1, o *i* pasako kelintą raidę tyrinėjame, kur, tarkime, A = 1, B = 2, ir t.t.

$$h_\theta^i(x) = P(y = i|x; \theta) \tag{6}$$

Remiantis 6 formule, apsibrėžiame tikimybės formulę, kuriai kaip parametrą paduodame gesto numerį (tarkime, kad raidė A = 1, B = 2 ir t.t.), bei bus naudojamasi parametrais θ, kur θ - svorių vertės, kuriomis remiantis apskaičiuojama tikimybė P. θ reikšmes savaime apsimokančios sistemos apsiskaičiuoja skirtingais algoritmais, todėl šiame darbe jie nagrinėjami nebus.

$$h_\theta(x) = g(z) \tag{7}$$

Remiantis 7 formule, galime apsibrėžti ne tik tiesės atskyrimą, kur gestas grąžins tikimybę didesnę už 0,5, kur ne, ne tik tiese atskirtus sprendinius, bet ir įvairias formas. Todėl tokiu atveju galima netgi apsibrėžti rankos formą. x_1 ir x_2 yra koordinačių ašys, kuriose yra išsidėstę taškai.

$$h_\theta(x) = \frac{1}{1 + e^{-z}} \tag{8}$$

Galiausiai, 8 formulėje galime matyti, jog tikimybė yra apskaičiuojama remiantis 7 formulėje apsibrėžtos formos pavidalu.

3.2.1.2. Linear Support Vector Machine

Iš esamų duomenų aibės taškų, šiuo atveju skirtingų gestų raidžių taškų, yra sudaroma matrica, kurioje tarp skirtingų gestų yra nubrėžiamas vektorius, kuris nusako, kurioje vietoje bus traktuojamas vienas gestas, kurioje - kitas. Iki vektoriaus yra parenkamas didžiausias galimas atstumas nuo

artimiausių prie vektoriaus esančių duomenų taškų.

Pats principas θ parinkimui yra labai panašus ir z yra paskaičiuojama pagal tą pačią formulę.

Šis metodas yra geresnis, kai yra ganėtinai didelis skirtumas tarp duomenų aibės taškų ir lengvai nubrėžiami vektoriai, nes nesikerta duomenų aibės. Dar vienas šio metodo privalumas yra tas, jog nubrėžus vektorių esant didžiausiam atstumui tarp duomenų taškų, esančių arčiausiai vektoriaus, ganėtinai nesunkiai yra nustatoma, kuriai pusei (šiuo atveju gestui), priklauso duotasis taškas.

Tačiau iš kitos pusės, šis metodas nėra ypač lengvai apdorojantis duomenis, jei duomenų aibės yra persidengiančios.

3.2.1.3. k-nearest neighbors

Viskas yra suskirstoma į aibes ir pagal tašką ieškomas artimiausias kaimynas arba kaimynų aibė ir priklausomai nuo to, kurių kaimynų daugiausia, pagal tai parenkama reikšmė.

Privalumai:

•

Trūkumai:

- Sudėtingumas ieškant artimiausių kaimynų kiekvienam taškui

3.2.2. Sukurtas modelis

4. Gestų kalbos statinių ženklų atpažinimas

4.1. Atpažinimas naudojant gestų nuotraukas

Norint, jog sistema apsimokytų kuo tiksliau, reikia jai duoti kuo daugiau duomenų. Sakykime, kad:

- a_{ik} - *i*-toji *k*-tosios abėcėlės raidės nuotrauka
- a_{1k}, a_{2k}, ..., a_{nk} - nuotraukų rinkinys, sudarytas iš *n* *k*-tosios raidės nuotraukų

Tuomet, turėdami $n * k$ nuotraukų, kuriuose vaizduojami gestų abėcėlės gestai, turime, jog tiek eilučių duomenų turėsime gestams atpažinti. Kuo gestai yra įvairesni prie skirtingų apšvietimų, skirtingų rankų ir pan., tuo tiksliau sistema pati galės vėliau atpažinti gestus.

4.2. Atpažinimas naudojant internetinę kamerą

Norint atpažinti statinius gestus naudojant internetinę kamerą, vienas iš to būdų yra naudotis *BackgroundSubtractorMOG2* klase. Ši klasė remiasi Gauso maišos priekinio plano/fono atskyrimo algoritmu

5. Gestų kalbos dinaminių ženklų atpažinimas

Išvados

Išvadose ir pasiūlymuose, nekartojant atskirų dalių apibendrinimų, suformuluojamos svarbiausios darbo išvados, rekomendacijos bei pasiūlymai.

Literatūra

[1] Christopher Olah. *Understanding LSTM Networks*. http://colah.github.io/posts/2015-08-Understanding-LSTMs/. 2015.

[2] Simon Haykin. *Neural Networks and Learning Machines*. 1 psl. Upper Saddle River, New Jersey 07458: Pearson Education inc., 2009.

[3] Tom Harrington. *ASL: Ranking and number of users*. http://libguides.gallaudet.edu/content.php?pid=114804&sid=991835/. 45 KB, atnaujinta 2016-07. 2004.

[4] World Health Organization. *Deafness and hearing loss*. http://www.who.int/mediacentre/factsheets/fs300/en/. 2017-02.