Универзитет у Београду – Електротехнички факултет

МАСТЕР РАД

на тему

Синтеза видео записа на основу говорног сигнала употребом рекурентних неуралних мрежа

кандидат:
Вељко Шешељ, 3253/2018

ментор:
доц. др Предраг Тадић

август, 2020.

Захвалница

Садржај

Увод

Компјутерска визија је комплексно поље које се бави процесирањем слике и видеа, односно она омогућава рачунарима да извуку информацију из видеа и слике. Постоји већ велики број технологија и апликација који користе компјутерску визију, попут аутономних возила, који користе анализу слике како би детектовали препреке и знакове на путу, медицинских система, који користе анализу слике како би се поставиле дијагнозе, и препознавања лица, које користе друштвене мреже како би предложили људе за означавање на фотографији. У комјутерску визију спада и *deepfake*, група програма који омогућавају синтезу лажних видеа (енг. *fake*) уз помоћ дубоких неуралних мрежа (енг. *deep neural networks*). Отуда потиче и назив *deepfake*. У овом раду ће бити представљен начин на који се може добити лахни видео особе која прича из аудио сигнала говора.

Добијање видеа из аудио снимка има веома широку примену. Особина добијања видеа високе резолуције из аудио снимка, значајно би смањила проток потребан за кодирање и трансмисију видеа. За људе са оштећеним слухом, техника би омогућила креирање видеа са кога би успешно могли да читају са усана. Такође, налазе велику примену у свету видео игара и филмских ефеката.

Историја *deepfake* програма започиње 1997. са радом [5], у коме је развијена иновативна техника која је могла да креира нове кратке видео анимације из аудио улаза, генерисаног из текста. Рад користи раније достигнућа за интерпретацију лица, генерисање аудиа из текста и $3D$ модела уста, али је први који је све методе обухватио заједно ради генерисања убедљивог лажног видеа. Уједно, ово истраживање представља и једно од најзначајнијих за развој *deepfake*-а. Током 2000., компјутерска визија је напредовала у пољу препознавања лица. Развој у овој области омогућио је боље праћење покрета лица, што је проузроковало убедљивије *deepfake* алгоритме. Једна таква метода је AAM алгоритам који статистички моделује облик лица [29].

Мана ранијих радова је што су захтевали субјекта у лабораторисјким условима где се контролишу услови попут осветљености лица, позе снимања и текста које субјект одговара, као и поседовање скупе рачунарске и видео опреме. За реализацију овог рада, битно је да је база података доступна на интернету и да се може реализовати уз помоћ релативно јефтине комерцијално доступне компјутерске опреме. Идеја представљена у раду [6] испуњава те услове. У раду се генерише лажни видео говора председника Барака Обаме из аудио снимка говора. На интернет сајту *YouTube* доступно је око 17 h видео материјала председничких обраћања нацији, снимљеног у периоду од 2008. до 2016. Видеи су јавно доступни у високој резолуцији и осветљење и позиција главе се не мењају драстично на њима. Упркос достпуности таквих података, генерисање лажног видеа је и даље тежак задатак, највише због осетљивости људског ока на промене у регији око уста током говора неке особе.

1 | Обележја говора

Први корак у синтези лажног видеа из говора је одређивање карактеристичних обележја говорног сигнала. Компонентне аудио сигнала треба да су довољно добре да носе лингвистичку компоненту, али и да су отпорне на позадински шум и на остале сметње. Једна од основних претпоставки везаних за обраду говорног сигнала јесте да се говор може приказати као излаз линеарног, временски променљивог система, чија се својства споро мењају са временом. То води ка основном принципу анализе говора који каже да ако се посматрају довољно кратки сегменти говорног сигнала, да се тада сваки сегмент може моделирати као излаз линеарног, временски инваријантног система [1]. Стога се кратки сегменти говора могу описати конволуционом једначином

$$s(t) = e(t) * \theta(t) \tag{1.1}$$

при чему $s(t)$ представља резултатни говорни сигнал, $e(t)$ представља побудну ваздушну струју (екситацију) и $\theta(t)$ импулсни одзив органа говорног тракта. У рачунарској обради говора, сигнале је згодније посматрати у дискретном домену

$$s[n] = e[n] * \theta[n] \tag{1.2}$$

Зависно од типа екситације, говорни гласови (фонеми) се могу поделити у три дистинктне категорије:

- звучни гласови
- фрикативи (безвучни гласови)
- плозиви

Код звучних гласова, ваздух прелази преко затегнутих гласних жица које почињу да вибрирају релаксираним осцилацијама, производећи квази-периодичне четвртке које ће побудити вокални тракт. Типични представници звучних гласова су самогласници. Побуда која производи фрикативе је окарактерисана широким спектралним садржајем као што је случајни шум (глас „ш”). Плозиви настају побудом која је високог интезитета и кратког трајања (попут Дираковог импулса) (гласови „б”, „п”, „т”...). Променом облика вокалног тракта мења се фреквенцијски садржај побуде и као резултат се добијају различити фонеми. Енглески језик разликује 42 фонема.

Проблем говорне анализе представља одређивање параметара екситације и параметара имплусног одзива вокалног тракта. Овај проблем се може назвати и проблемом раздвајања конволуционих компоненти, што је познато под називом деконволуција.

1.1 Кепструм

Кепструм [4] дискретног сигнала $s[n]$ се може израчунати помоћу формуле

$$c_s[n] = \boldsymbol{\mathcal{F}^{-1}}(\log(|\boldsymbol{\mathcal{F}}(s[n])|) \tag{1.3}$$

где су са $\boldsymbol{\mathcal{F}}$ и $\boldsymbol{\mathcal{F}^{-1}}$ означене дискретна Фуријеова трансформација(ДФТ) и инверзна ДФТ. Применом ДФТ на једначину 1.1 добија се

$$S(f) = E(f) \cdot \Theta(f) \tag{1.4}$$

Израчунавање кепструма може се сматрати системом за деконволуцију због чињенице да логаритам производа две компонетне представља збир логаритмованих компоненти

$$\log|S(f)| = \log|E(f) \cdot \Theta(f)| \tag{1.5}$$
$$= \log|E(f)| + \log|\Theta(f)| \tag{1.6}$$
$$= C_E(f) + C_\Theta(f) \tag{1.7}$$

Применом инверзне ДФТ добија се кепструм говорног сигнала

$$c_s[n] = \boldsymbol{\mathcal{F}^{-1}}(C_E(f) + C_\Theta(f)) \tag{1.8}$$
$$= \boldsymbol{\mathcal{F}^{-1}}(C_E(f)) + \boldsymbol{\mathcal{F}^{-1}}(C_\Theta(f)) \tag{1.9}$$
$$= c_e[n] + c_\theta[n] \tag{1.10}$$

У облику сигнала $c_s[n]$ уочљиве су области у којима доминирају еквиваленти ваздушне побуде $c_e[n]$, односно импулсног одзива говорних органа $c_\theta[n]$. Утицај ваздушне побуде доминантнији је при већим вредностима аргумента, док ниже вредности аргумента носе информацију о импулсном одзиву вокалног тракта. Због тога се најчешће користи првих 12 кепстралних коефицијената, док нулти коефицијент носи информацију о енергији сигнала.
Пошто је говор реалан сигнал, амплитуда спектра $|S(f)|$ је парна функција и кепструм ће имати реалне вредности

$$c_s[n] = \boldsymbol{\mathcal{F}^{-1}}(\log(|\boldsymbol{\mathcal{F}}(s[n])|) \tag{1.11}$$
$$= \frac{1}{N}\sum_{k=1}^{N-1} \log(|\boldsymbol{\mathcal{F}}(s[n])|) e^{\frac{j2kn\pi}{N}} \tag{1.12}$$
$$= \frac{1}{N}\sum_{k=1}^{N-1} \log(|\boldsymbol{\mathcal{F}}(s[n])|)(\cos\left(\frac{2kn\pi}{N}\right) + j\sin\left(\frac{2kn\pi}{N}\right)) \tag{1.13}$$
$$= \frac{1}{N}\sum_{k=1}^{N-1} \log(|\boldsymbol{\mathcal{F}}(s[n])|)\cos\left(\frac{2kn\pi}{N}\right) \tag{1.14}$$

Због овог својства, често се уместо инверзне ДФТ користи дискретна косинусна трансформација ради смањења комплексности израчунавања.

1.2 Мел-фреквенцијски кепстрални коефицијенти

Претходно описани поступци се често користе у применама везаним за аутоматско препознавање говора. У циљу опонашања људског начина доживљавања различитих учестаности у фонетима и примењујући кепстралну анализу настају мелфреквенцијски кепстрални коефицијенти (МФКК).

1.2.1 Мелова фреквенцијска скала

Мел-скала је усклађена са људским осећајем висине гласа односно његове учестаности. Њено добијање се врши експериментално, слушаоцу се репродукује тон учестаности 1000 Hz и као његово запажање о висини овог тона се бележи вредност 1000 mel, и ова вредност се користи као мера упоређивања за даље добијање мел скале. Затим се учестаност повећава све док слушалац не примети тон који слуша има дупло већу висину од упоредне вредности и та висина означава вредношћу од 2000 mel. Овај се принцип добија за добијање осталих вредности скале. Експерименти су ипак показали да је међусобна зависност мела и херца лиенарна до 500 Hz, док изнад ове учестаности једнаким променама мела одговара све већа промена у херцима. За конвертовање фреквенције у мелову скалу и назад могу се користити формуле

$$M(f) = 1125 \log\left(1 + \frac{f}{700}\right) \tag{1.15}$$

$$M^{-1}(m) = 700\left(\exp\left(\frac{m}{1125}\right) - 1\right) \tag{1.16}$$

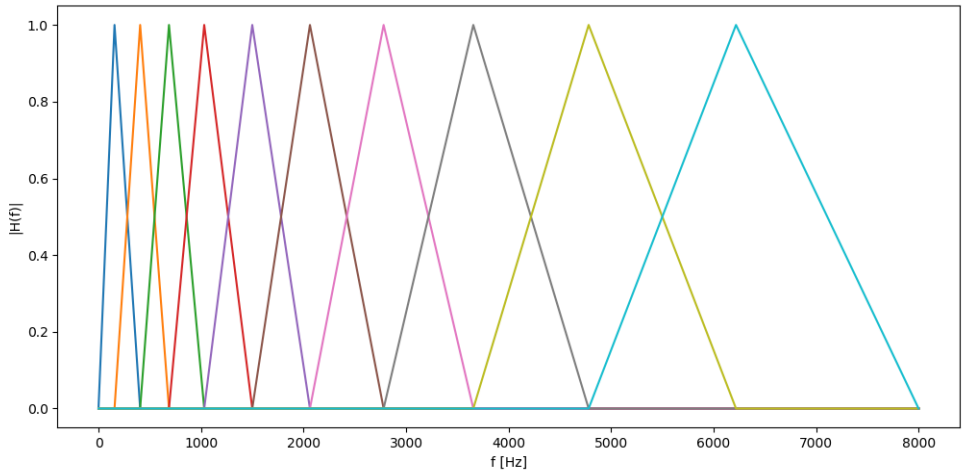
1.2.2 Мелова банка филтара

Постојање аудиторних критичних опсега је такође особина која условљава људски доживљај различитих учестаности. Ова појава везана је за чујни доживљај слушаоца приликом слушања два различита тона на различитим учестаностима. Дакле, уколико се слушоцу пусти тон учестаности f_1 која се налази у неком чујном опсегу, тада слушалац има доживљај тона у складу са мел склаом. Уколико се поред тог тона пусти други тон учестаности f_2 тада звучни доживљај слушаоца зависи од међусобне фреквентне блискости ова два тона. Наиме, уколико је фреквентни размак довољно мали тако да се налазе унутар истог критичног чујног опсега долази до појаве маскирања и слушалац чује тон f_1, али веће гласности. Уколико је фреквентни размак ових тонова већи од ширине чујног критичног опсега тада слушалац чује два тона на одговарајућим претходно поменутим учестаностима. Имајући то у виду фреквенцијски опсег говорног сигнала потребно је изделити на опсеге. Ти опсези формирају мелову банку филтара. Поступак формитања мелове банке филтара се може изделити на кораке:

1. Користећи јендачину 1.15 конвертовати доњу и горњу границу говорног сигнала у мелову скалу. За доњу границу се може узети вредност од 0 Hz, док је горња фреквенција обично условљена Никвистовим креитеријумом. У коришћеном скупу података, говрни сигнал је одабиран са 16 kHz, па је за горњу границу коришћена вредност од 8 kHz.

2. Изделити опсег говорног сигнала на меловој скали на еквидистантне делове (опсеге). За број делова се обично узима број из интервала [26, 40]. За n филтара добијају се $n + 2$ фреквенције на меловој скали.

3. Добијене мелове фреквенције, потребно је вратити у Hz фреквенције, формулом 1.16, које се касније користе за централне учестаности троугаоних филтара



Слика 1.1: Говорни фреквенцијски опсег подељен на 10 мел филтара

1.2.3 Алгоритам израчунавања МФКК

У овом поглављу ће бити приказан детаљан поступак добијања МФКК.

- **Високофреквентно филтрирање**
  Говорни сигнал је по природи аналогни сигнал, који је потребно дискретизовати и дигитализовати да би се израчунала тражена обележја. Поменути процеси су нискофреквентни и утичу на слабљење виших спектралних компоненти у говорног сигналу. Из тог разлога након извршене дигитализације, а пре самог издвајања обележја, потребно је извршити предобраду снимљеног говорног сигнала. То се постиже применом високопропусног филтра првог реда
  $$H(z) = 1 - az^{-1} \tag{1.17}$$
  при чему се параметар a бира из интервала $[0.95,\ 0.98]$[4].

- **Прозоровање сигнала**
  На почетку поглавља је речено да се само кратки сегменти говора могу сматрати као излаз линеарног, временски инваријантног система. Са тим у вези сигнал је потребно изделити на прозоре дужине 20 ms-40 ms [1]. У овом раду изабрана је дужина прозора од 25 ms, са преклапањем између сусдених прозора од 15 ms. Наредни кораци се примењују за сваки прозор.

- **Израчунавање спектра снаге**
  За сваки прозор говорног сигнала је првобитно потребно одредити ДФТ. Ради потискивања бочних лобова, згодно је применити Хамингов прозор пре израчунавања ДФТ. У овом раду, ДФТ за сваки простор се израчунава у 512 тачака. Спектар снаге се може естимирати периодограмом, који за сигнал $s[n]$ се израчунава по формули
  $$S(f) = \boldsymbol{\mathcal{F}}(s[n]) \tag{1.18}$$
  $$P(f) = \frac{1}{N}|S(f)|^2 \tag{1.19}$$

- **Филтрирање спектра снаге меловом банком филтара**
  Добијени периодограм је потребно филтрирати кроз сваки филтар из мелове банке. Добијени коефицијенти за сваки филтар се сумирају. На крају се добијају коефицијенти који нам говоре колико је енергије садржано у сваком филтру филтер банке.

- **Логаритмовање**
  Логаритмовати сваки коефицијент добијен пропуштањем периодограма кроз банку филтара.

- **Дискретна косинусна трансформација**
  Дискретном косинусном трансформацијом над логаритмованим енергијама добијају се кепстрални коефицијенти.

1.2.4 Динамичка обележја

Често је од интереса посматрати и промену МФКК у времену [4]. На овај начхин, коришћена обележја директно носе информацију о променама између суседних прозора говорног сигнала. Ради израчунавања ових обележја користе се полиномијалне апроксимације првог и другог извода кепстралних коефицијената [7]

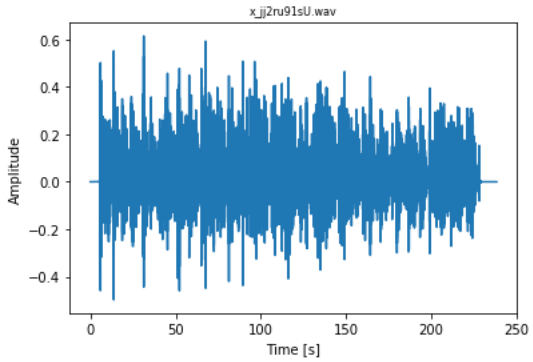
$$\Delta c_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{1.20}$$

$$\Delta^2 c_t = \frac{\sum_{n=1}^{N} n(\Delta_{t+n} - \Delta_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{1.21}$$

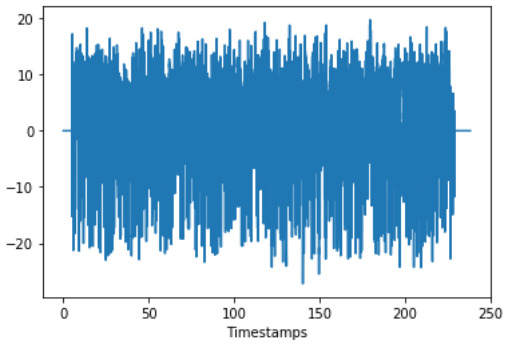
где су Δc_t и $\Delta^2 c_t$ делта и делта-делта кепстрални коефицијенти за прозор t, израчунати из статичких кепстралних коефицијената из околних прозора. Типична вредност која се узима је $N = 2$.

1.3 Приказ обележја говора

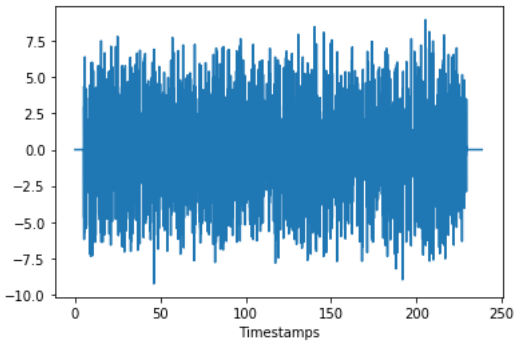
У овом раду, као вектор улазних параметара, коришћено је 13 кепстралних коефицијената, са њихом делтама, што даје улазни вектор димензије 26. Уместо првог кепстра, коришћена је енергија сигнала. Из базе видео фајлова, само аудио снимци су извучени уз помоћ алата *FFmpeg* [8], чиме се добија база аудио снимака говора председника Обаме. Након тога, сваки аудио фајл је нормализован уз помоћ $FFmpeg - normalize$ додатка [9]. Приказ неких кепстралних коефицијената, за насумично одабран аудио снимак из базе, се може погледати у наставку.



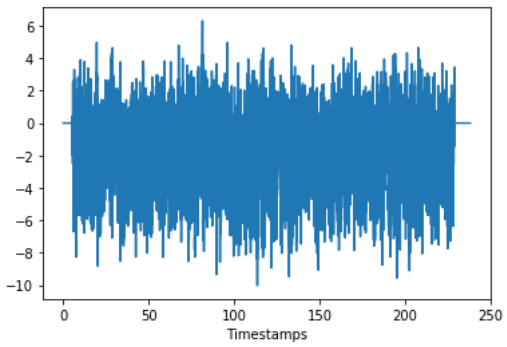
Слика 1.2: Нормализовани аудио сигнал насумично изабран из базе



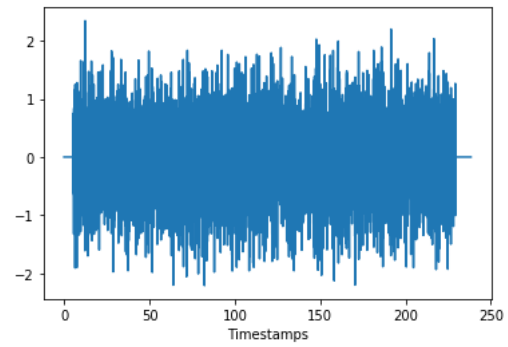
(а) Први МФКК



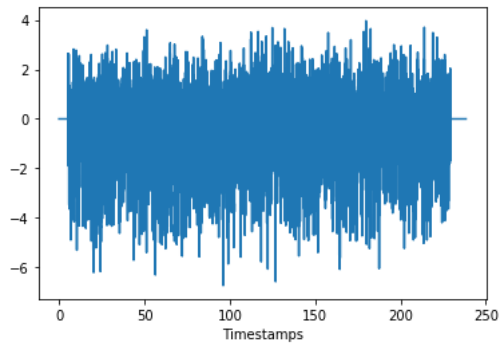
(б) Делта првог МФКК



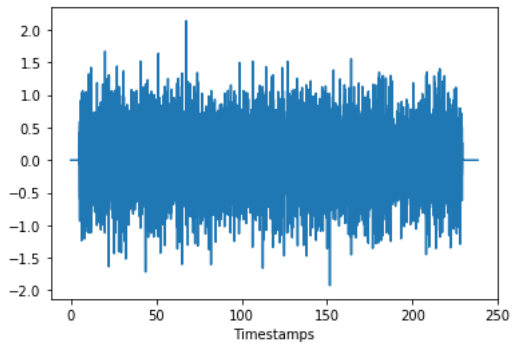
(ц) Шести МФКК



(д) Делта шестог МФКК



(е) Дванаести МФКК



(ф) Делта дванаестог МФКК

Слика 1.3: Приказ гласовних обележја

2 | Обележја видеа

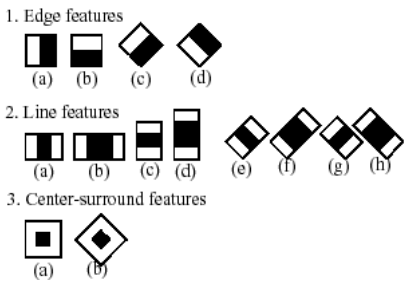
У овом поглављу ће бити објашњен начин на који се из видео снимака може добити база која ће садржати координатне тачака уста. Процес се може поделити на три дела: детектовање лица говорника на фрејму видеа, детектовање координата 68 значајних тачака лица, прављење $3D$ модела лица из ког ће бити издвајене $3D$ координате појединих тачака уста.

2.1 Детекција лица

Детекција и препознавање лица је једна од најпопуларнијих тема компјутерске визије у последњих неколико година. Ова технологија је широко заступљена, од камера које фокусирају лица пре фотографисања, до друштвене мреже *Facebook* која препознаје индетитет корисника на сликама. Комјутерски програм који одлучује да ли је слика позитивна, односно да ли постоји лице на њој, или негативна, односно да не постоји, се назива класификатор. Класификатор је обучен на стотинама хиљада позитивних и негативних слика, како би се нова слика класификовала исправно. Библиотека *OpenCV* [25] нуди два већ обучена класификатора

- Харов[1] клафикатор
- *LBP* класификатор

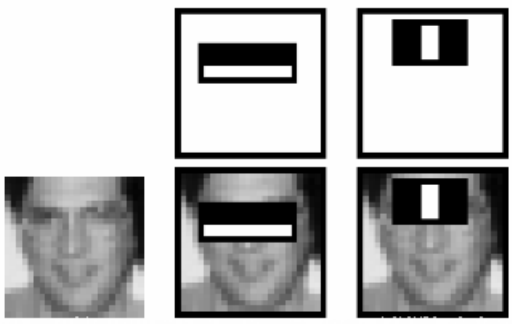
У овом раду је коришћен Харов класификатор, који представља ефективни метод детекције објеката уз помоћ обележја и представљен је у раду [26]. Иницијативно, алгоритам креће од велике базе података позитивних и негативних слика над којима екстрактује Харова обележја. Да би се израчунала Харова обележја, на сваку слику се примењују конволуциони кернели са слике 2.1. Свако обележје се рачуна као разлика суме вредности пиксела испод белог правоугаоника и суме вредности пиксела испод црног правоугаоника.



Слика 2.1: Конволуциони кернели за израчунавање Харових обележја

[1]Алфред Хар (1885. - 1933.)- мађарски математичар

Смисао методе се може показати на примеру са слике 2.2 где се екстрактују два обележја. Први се базира на појави да је регион око очију често тамнији од регије носа и образа. Друго облежје се базира на томе што су очи тамније од ивице носа.



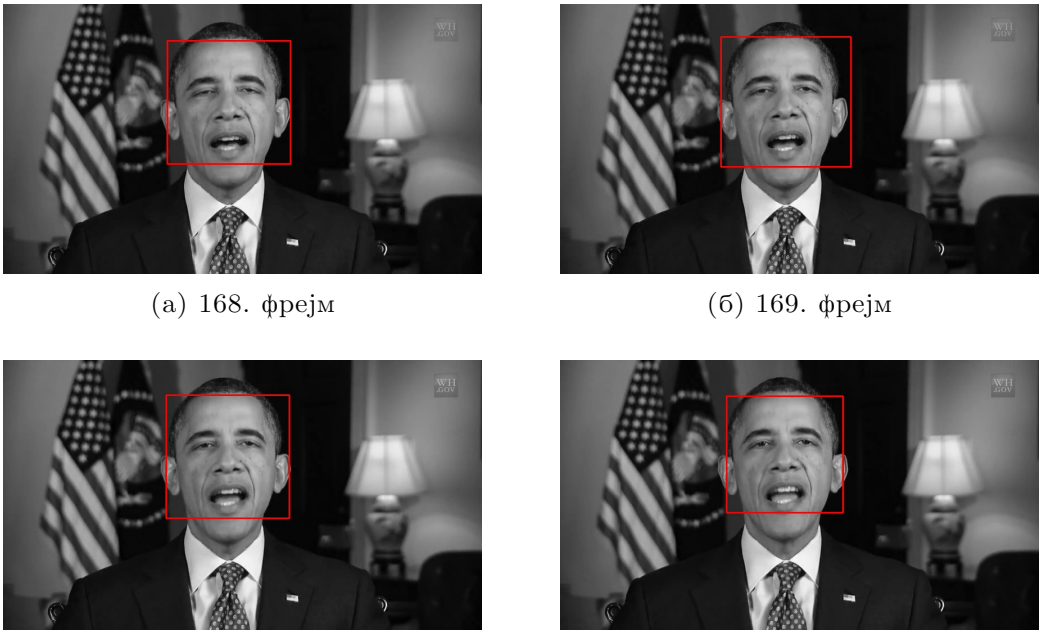
Слика 2.2: Приказ израчунавања два Харова обележја

За рачунање свих обележја једне слике користе се сви могући кернели, у свим могућим величинама на сваком делу слике, што резултује великим бројем обележја. Већина облежја израчуната оваквом методом је ирелевантна. Алгоритам који се користи за обучавање детектора лица и који има ту могућност да разматра само најбитнија обележја је *Adaboost*. У својој оригиналној форми, *AdaBoost* алгоритам обучавања се користи како би побољшао класификацију једноставних (слабих) алгоритама обучавања. За свако обележје, нађе се најбољи праг који ће класификовати слике на позитивне и негативне (пањ одлучивања као слаби ученик [3]). За рачунање обележја на сликама, кернел је фиксне димензије 24×24. У свакој итерацији бирају се пањеви са минималном грешком, повећавају се тежински коефицијенти погрешно класификованих слика, смањују се тежински коефицијенти тачно класификованих слика и рачуна се тежина слабог ученика. Након одређеног броја итерација, крајњи класификатор представља тежинску суму слабих ученика. На крају је добијено око 6000 обележја.

Како би се даље оптимизовало и унапредило израчунавање обележјана слици, уводи се појам каскадног класификатора. Каскадни класификатор се састоји од неколико нивоа, где се сваки ниво састоји од неколико пањева одлучивања. Сваки ниво означава регион слике, одређен тренутном позицијом померајућег прозора, као позитиван или негативан. У случају да је регион негативан, класификатор прелази на наредну локацију, а ако је регион позитиван, регион прелази на наредни ниво. Детектор пријављује да је лице у региону детектовано у случају да је регион прошао све нивое. Нивои су дизајнирани тако да одбаце негативне регионе што брже, због претпоставке да већина региона не садржи објекат од интереса, што је у овом случају лице. Да би метод добро радио, сваки ниво мора да има низак број лажно негативних детекција. Ако ниво не детектује лице на региону где лице постоји, каскадна класификација се зауставља и грешка се не може отклонити. Међутим, сваки ниво сме имати висок број лажно позитивних детекција, јер уколико детектор детектује лице на региону који не садржи лице, грешка се може уклонити у наредним фазама. Додавање фаза смањује број лажно позитивних детекција, али и смањује број истински позитивних детекција.

Овако обучен класификатор доступан је уз *OpenCV* библиотеку. У раду је одлучено да се користи Харов класификатор, будући да има већу прецизност при детекцији и умањен број лажно позитивних детекција, у односу на други доступан

LBP класификатор. Мана Харовог класификатора је што је комплексан и спор за израчунавање и мање прецизан на тамнопутим лицима. Будући да се ради о председнику Обами, друга ставка је представљала потенцијалан проблем, али није примећен неки значајан број фрејмова на којима лице није детектовано. Током видеа недељних обраћања нацији, дешава се да камерман неколико пута мења позу и осветљење којим снима говорника, у колико се у неком делу није детектовало лице тај део видеа би био прескочен. Такође, на видеима је присутно само једно лице, и у случају детекције више лица, бирало би се оно које је најближе лицу које је детектовано у претходном фрејму.



(а) 168. фрејм (б) 169. фрејм

(ц) 170. фрејм (д) 171. фрејм

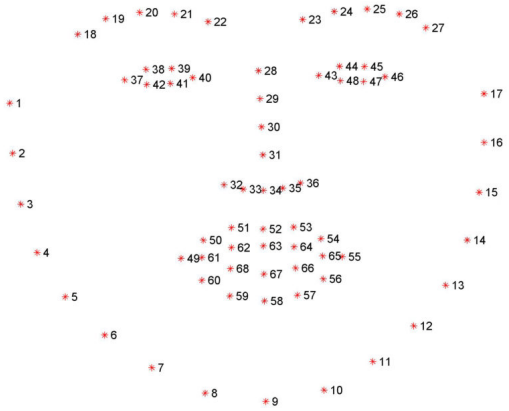
Слика 2.3: Приказ детектованих лица над фрејмовима видеа *Weekly Address*: *A Balanced Approach to Growing the Economy in* 2013

2.2 Детекција карактеристичних тачака лица

Детекција карактеристичних тачака лица (енг. *facial landmarks*) је користан алгоритам потребан у различитим апликацијама као што су препознавање лица, препознавање емоција на лицима, замена лица, препознавање положаја главе, филтери на апликацијама *Instagram* и *Snapchat*... Кракатеристичне тачке се користе како би се локализовале и представиле видљиве регије лица као што су очи, обрве, нос, уста и вилица. Детекција таквих тачака представља подскуп проблема у којима циљ предикција неког облика. Предиктор облика има за циљ да на регији од интереса слике локализује кључне тачке дуж тог облика. Стога се цео процес може поделити на два дела, први који ће детектовати лице (поглавље 2.1) и детекција кључних тачака.

Постоји велики број расположивих детектора карактеристичних тачака лица, али већина њих се труди да што боље детектује следеће значајне регије: уста, десна обрва, лева обрва, десно око, лево око, нос и вилица. Детектор који је доступан уз библиотеку *OpenCV* и који је коришћен у овом раду је *LBF* модел (енг. *Local*

Binary Features, представљен у раду [27]. Облик лица $S = [x_1, y_1, ..., x_{N_{fp}}, y_{N_{fp}}]^T$ се састоји од N_{fp} карактеристичних тачака. За дату слику, циљ естимације може бити облик S који ће бити што приближнији истинитом облику $\hat{S}$ који ће минимизовати $||S - \hat{S}||$. Ова критеријумска функција се најчешће користи у току тренирања, како би се проценио перформанс. Аутори рада [27] користе алгоритам случајне шуме како би одредили функцију мапирања тачака на слици. У зависности која је база података коришћена при тренирању, излаз модела ће имати другачији број карактеристичних тачака на излазу. За *LBF* модел, коришћен је *iBUG* 300*W* сет, који садржи слике са ручно означеним 68 тачака.



Слика 2.4: Приказ 68 карактеристичних тачака лица

Од значаја је приметити да тачке које представљају уста имају индексе од 49 до 68. Резултате детекције можете видети на наредним сликама.

(а) 168. фрејм

(б) 169. фрејм

(ц) 170. фрејм

(д) 171. фрејм

Слика 2.5: Приказ детектованих карактеристичних тачака лица над фрејмовима видеа *Weekly Address*: *A Balanced Approach to Growing the Economy in* 2013

2.3 3D модел лица

$3D$ морфолошки модели (енг. $3D$ *Morphable Model-* $3DMM$) су моћан алат у компјутерској визији. Они имају апликацију у задацима где се захтева $2D$ процесирање лица као што су, анализа лица, препознавање, естимација и нормализација позе [2]. $3DMM$ су први пут предложене у раду [28] и од тада се примењују у различитим задацима. Међутим, и даље нису толико заступљени као $2D$ методе, попут активног модела излгеда [29] (енг. *Active Appearance Model- AAM*), иако имају одређене предности уодносу на њих. У $3D$ моделу, поза лица је одвојена од облика. Његова пројекција у $2D$ је моделована од стране модела физичке камере. Камера модел представља функцију која мапира $3D$ простор у простор слике. Такође, коришћење $3D$ модела омогућава експлицитно моделовање извора светла, јер су информације о површини и дубини објекта познате. Модел извора сетла одваја светлост од изгледа лица и тиме јачина светлости не утиче на параметре текстуре, као што је случај у AAM моделима. Даље, $3DMM$ могу да се користе како би се генерисала специфична лица, или како би се генерисали подаци за друге алгоритме, јер покривају различите позе, укључујући и екстремне као што су погледи из профила. Ово последње може бити јако значајно, у случају да се током видеа говора, мењају углови из којих је говорник сниман на такав начин да део лица није видљив. Са друге стране, $3DMM$ су тешки за тренирање и захтевају доста рачунарске снаге за израчунавање.

Један $3DMM$ који је доступан на интернету је Сари [3] $3DMM$ модел лица, који је представљен у раду [30]. Модел се састоји од три нивоа резолуције, од који је само најнижи доступан за некомерцијалне сврхе. Уз модел, омогућена је $C++$ библиотека која олакшава фитовање позе и облика на новим сликама (фрејмовима) [31].

Модел се састоји од два PCA модела [4]: PCA модел облика и PCA модел боје (текстуре). Основна идеја PCA методе, односно методе Кархунен-Лоеве експанзије [2], је да редукује димензију вектора обележја ради упрошћавања модела, а да губитак информација и прецизности буде минималан. Матода сматра да се информација крије у координатама великих распиања, односно варијанси. Лице се може представити као вектор $S \in \mathbb{R}^{3N}$, који садржи x, y и z компоненте облика (вертексе), и вектора $T \in \mathbb{R}^{3N}$ који носи информацију о RGB боји сваког вертекса. Сваки PCA модел $\mathbf{M}$

$$\mathbf{M} = (\bar{\mathbf{v}}, \sigma, \mathbf{V}) \tag{2.1}$$

се састоји од компоненти $\bar{\mathbf{v}} \in \mathbb{R}^{3N}$, који представљају средњу вредност $3D$ полигоне мреже (енг. *polygonal mesh*), скупа принципијалних компоненти $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_{n-1}] \in \mathbb{R}^{3N\times(n-1)}$ и стандардне девијације $\sigma \in \mathbb{R}^{n-1}$, где је n број $3D$ скенова који су коришћени за прављење модела. Нова лица се могу генерисати израчунавањем

$$\mathbf{S} = \bar{\mathbf{v}} + \sum_{i}^{M} \alpha_i \sigma_i \mathbf{v}_i \tag{2.2}$$

за облик, где је $M \leq n-1$ је број принципјалних компоненти и $\alpha in\mathbb{R}^M$ су $3D$ координате нове инстанце у PCA простору. Сличан метод се може применити и за

[2] Естимација позе представља задатак да се на основу $3D$ модела и његове слике, $2D$ пројекције, добију транслација и ротација објекта такви да дају ту $2D$ пројекцију

[3] *University of Surrey*-универзитет из Велике Британије

[4] енг. *PCA- Principal Component Analysis*

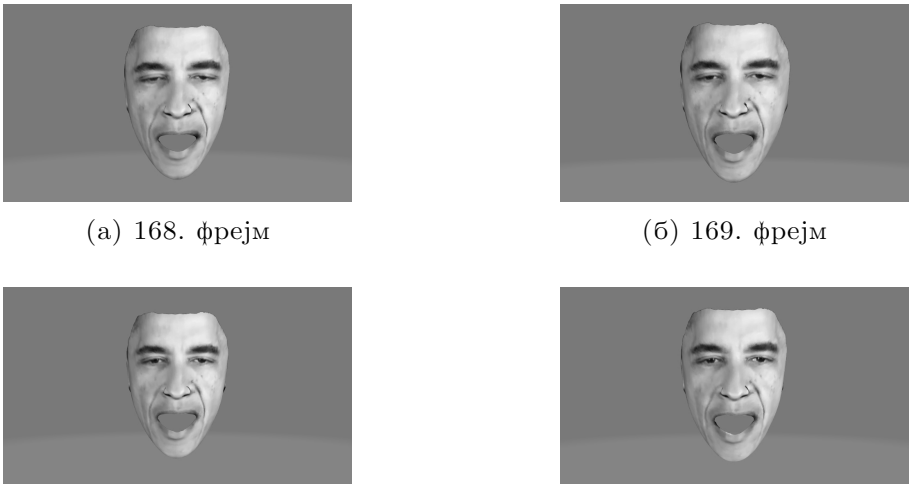
боју (албедо).
SFM је изграђен на великом броју $3D$ скенова високе резолуције лица људи разлхите старосне доби и разлхите боје коже. Важан податак је да се у самом сету најмање најмање тамнопутих лица (око 5%), али због непоседовања других $3DMM$ модела, одлучено је да се остане при SFM.

Слика 2.6: Процес добијања једног $3D$ скена за базу података

Из скенова се формарају вектори вертекса и вектори боја. У зависности од нивоа резолуције модела, ти вектори су различите димензије. Из вектора вертекса и боја рачунају се матрице коваријансе из којих се чувају сопствени вектори: 63 за облик и 132 за боју.. Расположива библиотека укључује методе потребне за фитовање позе, облика и могућност фронтализације лица. За дати сет $2D$ карактеристичних тачака лица потребо је знати у које тачно вертексе модел камере слика те тачке, као што је то случај за широко заступљене $iBUG$ 68 карактеристичних тачака. У том случају, добијање модела камере се своди на решавање линеарног система једначина. $3D$ координате облика у PCA простору α се могу наћи минимизовањем

$$\mathbb{E} = \sum_{i}^{3N} \frac{(y_{m2D,i} - y_i)^2}{2\sigma_{2D}^2} + ||\alpha|| \tag{2.3}$$

где је N број карактеристичних тачака, y су детектоване или означене карактеристичне тачке, σ_{2D}^2 је опциона варијанса ових карактеристихних тачака и y_{m2D} је пројекција од $3DMM$ облика у $2D$ добијена коришћењем естимираног модела камере. Слично се може применити и за модел текстуре.



(а) 168. фрејм (б) 169. фрејм

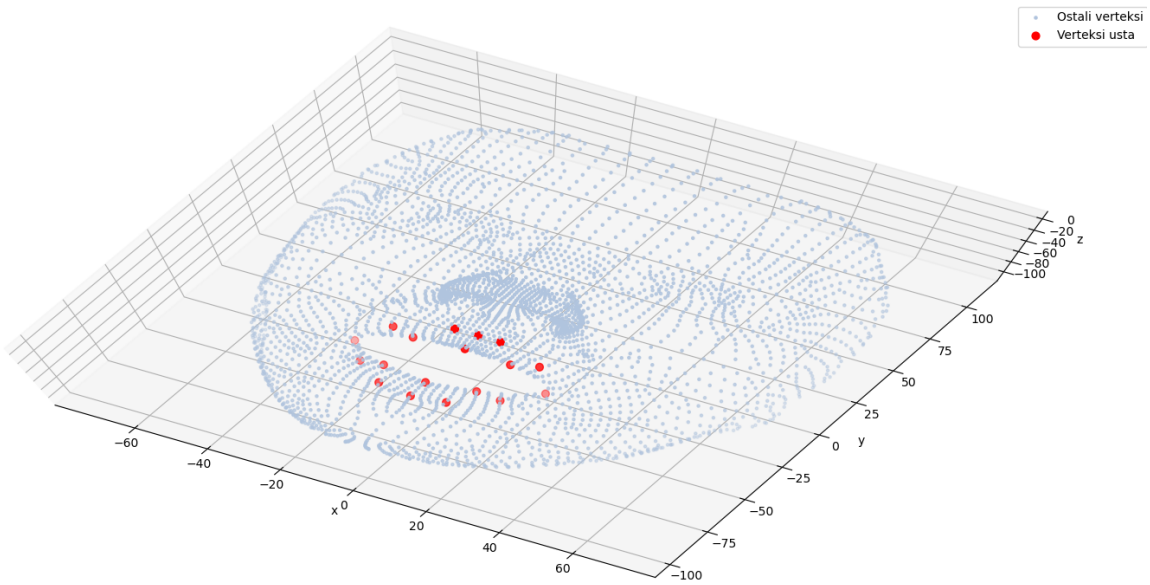
(ц) 170. фрејм (д) 171. фрејм

Слика 2.7: Приказ генерисаних $3D$ модела лица из фрејмовима видеа *Weekly Address*: *A Balanced Approach to Growing the Economy in* 2013

Коришћење дате библиотеке захтева доста процерског времена. Уз помоћ компајлерских оптимизација доступним у оквиру програмског језика $C++$, смањивањем резолуције видеа на $480p$, 17 h видео материјала је обрађено за две недеље. Добијен $3D$ модел је записан у формату *.obj*, који представља формат који се може отварати у програмима попут $3D$ *Paint*. Формат *.obj* је читљив текстуални фајл [32] који садржи листу вертекса са њиховим координатама, листу полигоналних облика које ти вертекси формирају, као и информације о текстури сваке полигоналне површи. Текстура се чинформацију о томе у које се вертексе слика 68 карактеристичних тачака. Бирајући само оне вертексе који представљају $3D$ пројекцију тачака који припадају региону уста, добија се 17 тачака са три координате, што даје излазни вектор димензије 54.

Индекс тачке међу 68 карактеристичних тачака	Индекс вертекса у $3D$ моделу
49	398
50	315
51	413
52	329
53	825
54	736
55	812
56	841
57	693
58	411
59	264
60	431
61	није дефинисан
62	416
63	423
64	828
65	није дефинисан
66	817
67	442
68	404

Табела 2.1: Мапирање карактеристичних тачака уста на $3D$ модел



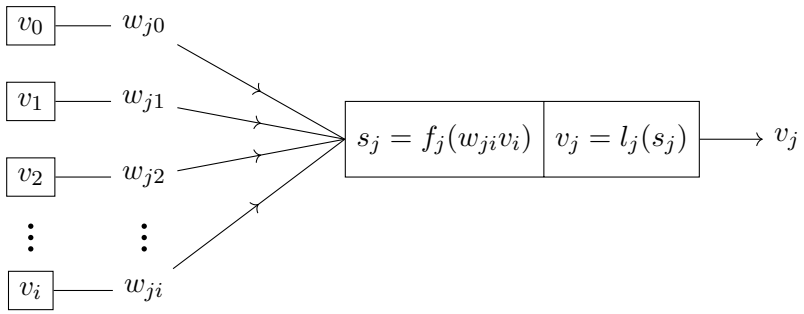
Слика 2.8: Приказ одабраних вертекса

3 | Рекурентне неуралне мреже

Неуралне мреже су модели учења који остварају огроман успех у широком спектру задатака супервизираног и несупервизираног машинског учења. Класичне (*feedforward*) неуралне мреже се посебно издвајају у проблемима класификације. Упркос њиховој моћи, стандардне неуралне мреже имају ограничења, од којих је назначајније да примери из базе података морају бити међусобно независни. У случају када су подаци зависни у времену и простору ово није прихватљиво. Фрејмови видеа, делови аудио сигнала, речи из реченице представљају примере података где захтев о међусобној независности није задовољен. Рекурентне неуралне мреже превазилазе тај недостатак и користе се у случахевима када се подаци могу приказати у форми секвенце. Оне имају особину да селективно прослеђују информацију између корака секвенце, док и даље процесују један по један корак. У овом поглављу ће се прво бацити кратак осврт на класичне (*feedforward*) неуралне мреже, а затим ће бити детаљније обрађен појам различитих врста рекурентних неуралних мрежа.

3.1 Преглед вештачких неуралних мрежа

Неуралне мреже су модели израчунавања инспирасни биолошким неуралним системом. Неурална мрежа се састоји од скупа вештачких неурона, који се још и називају чворовима, који су међусобно повезани синапсама (везама). За сваки неурон j се дефинише функција активације $l_j(*)$ и интеграциона функција $f_j(*)$. Веза од чвора j' ка чвору j је описана тежинским коефицијентом $w_{jj'}$.

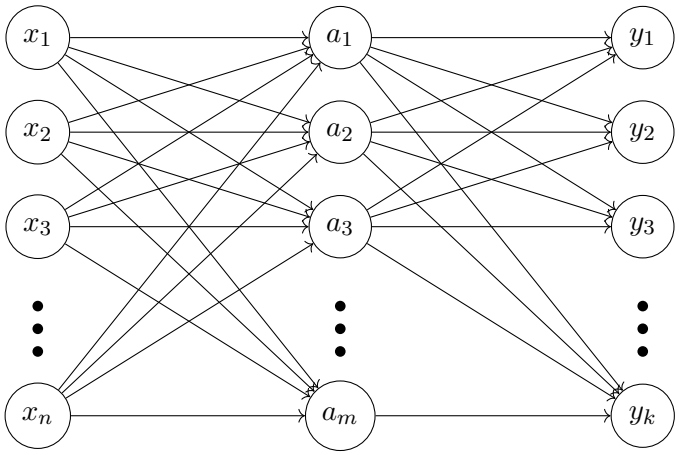


Слика 3.1: Шематски приказ вештачког неурона

За интеграциону функцију се најчешће узима линеарна сума, те се вредност на излазу једног чвора може израчунати као

$$v_j = l_j\left(\sum_{k=0}^{i} w_{ji}v_i\right) \tag{3.1}$$

Чести избори активационе фунцкије су сигмоид $\sigma(z) = 1/(1 + e^{-z})$ и хиперболички тангенс $\tanh(z) = (e^z - e^{-z})/(e^z - e^{-z})$. Од скора, у моделима дубоких неуралних мрежа користи се активациона функција $ReLU(z) = max(0, z)$ која је показала значајно побољшање перформанси.
Feedforward неурлне мреже су врста вештачких неуралних мрежа чији усмерени графови не смеју да садрже петље. Због одсуства петљи, чворове мреже је могуће организовати у слојеве. Излаз једног слоја се добија на основу излаза нижих слојева.



Слика 3.2: Шемтаски приказ вишеслојне *feedforward* неуралне мреже

Вектор обележја $\mathbf{x}$ се доводи на најнижи (улазни) слој. Излази чворова у наредним слојевима се сукцесивно израчунавају, док се не добије излаз највишег (излазног) слоја мреже $\hat{\mathbf{y}}$. *Feedforward* мреже се користе у супервизованом учењу на задацима класификације и регресије. Учење се остварује ажурирањем тежинских коефицијената $w_{jj'}$ са циљем да се оптимизује критеријумска функција $\mathfrak{L}(\hat{\mathbf{y}}, \mathbf{y})$, која пенализује дистанцу између излазног вектора $\hat{\mathbf{y}}$ и вектора циља $\mathbf{y}$.
Најуспешнији алгоритам за транирање неуралних мрежа је алгоритам пропагације у назад (енг. *backpropagation*)[10]. Алгоритам пропагације у назад користи ланчано правило за рачунање извода критеријумске функције $\mathfrak{L}$. Тежински коефицијенти се поправљају користећи алгоритам градијентног спуста. Најчешће коришћен алгоритам градијентног спуста је градијентни спуст који користи мини шарже (енг. *batch*) обучавајућег скупа. Тај алгоритам комбинује предности шаржног и стохастичког градијентног спуста. Применом тог алгоритма коефицијенти се ажурирају на основу акумулиране грешке свих примерака из мини шарже. За шаржу дужине n правило ажурирања тежинских коефицијената се може записати овако

$$w := w - \eta \nabla_w \mathfrak{L}\left(\hat{\mathbf{y}}^{\mathbf{i:i+n}}, \mathbf{y}^{\mathbf{i:i+n}}\right) \tag{3.2}$$

где је са η означена брзина обучавања (енг. *learning rate*). У опстем случају, критеријумска функција није конвексна, и не постоји гарант да ће градијентни спуст довести до глобалног минимума. Многе варијанте градијентног спуста се уводе како би се убрзало обучавање. Неке од најпопуларнијих су: *AdaDelta* [11], *AdaGrad*

[12], *RMSprop* [13] и *Adam* [14]. Углавном се заснивају на мењању брзине обучавања као и на моментуму који представља промену тежинског коефицијента у претходној итерацији алгоритма.
Да би се израчунао градијент у једначини 3.2 користи се, већ поменут, алгоритам пропагације у назад. Као што јој само име каже, алгоритам започиње од излазног слоја и креће уназад ка нижим слојевима. За чвор j, који се налази у излазном слоју, активационе функције $l_j(*)$ и линеарном интеграционом фунцијом $s_j(*)$, важи једначина 3.1

$$y_j = v_j = l_j\left(s(w_{ji}, v_i)\right) \tag{3.3}$$

$$= l_j\left(\sum_{k=0}^{i} w_{ji} v_i\right) \tag{3.4}$$

За тежински коефицијент везе између чвора j из излазног слоја и цвора i из скривеног слоја, промена се може добити на следећи начин

$$\Delta w_{ji} = -\eta \frac{\partial \mathfrak{L}\left(\hat{\mathbf{y}}, \mathbf{y}\right)}{\partial w_{ji}} \tag{3.5}$$

$$= -\eta \left[\frac{\partial \mathfrak{L}\left(\hat{\mathbf{y}}, \mathbf{y}\right)}{\partial y_j}\right] \left[\frac{\partial y_j}{\partial s(w_{ji}, v_i)}\right] \left[\frac{\partial s(w_{ji}, v_i)}{\partial w_{ji}}\right] \tag{3.6}$$

$$= \eta \delta_{ji} v_h \tag{3.7}$$

$$\delta_{ji} = -\left[\frac{\partial \mathfrak{L}\left(\hat{\mathbf{y}}, \mathbf{y}\right)}{\partial y_j}\right] \left[\frac{\partial y_j}{\partial s(w_{ji}, v_i)}\right] \tag{3.8}$$

$$= -\left[\frac{\partial \mathfrak{L}\left(\hat{\mathbf{y}}, \mathbf{y}\right)}{\partial y_j}\right] l_j'(s_j) \tag{3.9}$$

Величина δ_{ji} се назива и сигналом грешке. Први чинилац у изразу 3.8 зависи од избора критеријумске функције, а други представља извод активационе функције $l_j'(*)$. За тежински коефицијент везе између чвора који припада скривеном слоју i и чвора који се налази у слоју иза l важи

$$\Delta w_{il} = -\eta \frac{\partial \mathfrak{L}\left(\hat{\mathbf{y}}, \mathbf{y}\right)}{\partial w_{il}} \tag{3.10}$$

$$= -\eta \left[\frac{\partial \mathfrak{L}\left(\hat{\mathbf{y}}, \mathbf{y}\right)}{\partial v_i}\right] \left[\frac{\partial v_i}{\partial s_l}\right] \left[\frac{\partial s_l}{\partial w_{il}}\right] \tag{3.11}$$

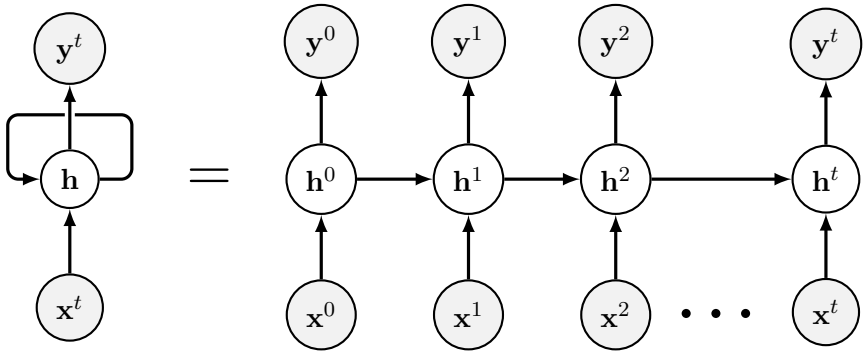
$$= \eta \delta_{il} v_l \tag{3.12}$$

$$\delta_{il} = l_l'(s_l) \sum_k \delta_{kl} w_{kl} \tag{3.13}$$

Једначина 3.13 се сукцесивно примењује за све ниже слојеве, чиме се уз сачуване вредности излаза чворова добијају градијенти појединачних тежинских коефицијената.

3.2 Рани модели рекурентних неуралних мрежа

Рекурентне неуралне мреже су мреже чији усмерени графови дозовљавају петље, чиме се уводи појам времена у модел. У тренутку t, чворови са рекурентним везама зависе од тренутног улазног вектора $\mathbf{x}^{(t)}$ и вредности стања чвора у претходном тренутку $\mathbf{h}^{(t-1)}$. Излаз чвора $\hat{\mathbf{y}}^{(\mathbf{t})}$, зависи од вредности стања чвора $\mathbf{h}^{(t)}$ и посредством рекурентних веза улаз улаз у тренутку $t-1$ $\mathbf{x}^{(t-1)}$ може утицати на излаз у тренутку t.



Слика 3.3: Шематски приказ одмотане рекурентне неуралне мреже у времену

Мрежа са слике 3.3 се може описати једначинама

$$\mathbf{h}^t = l_h(\mathbf{W_{hx}}\mathbf{x}^t + \mathbf{W_{hh}}\mathbf{h}^{t-1} + \mathbf{b_h}) \tag{3.14}$$

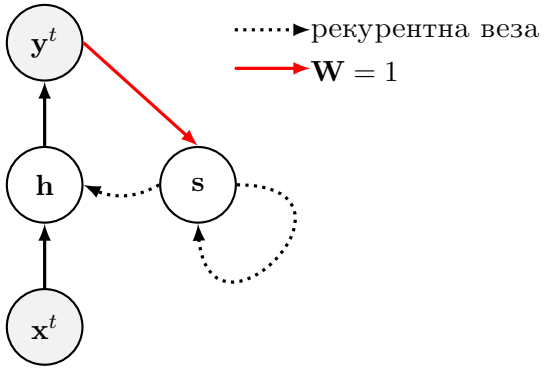
$$\mathbf{y}^t = l_y(\mathbf{W_{yh}}\mathbf{h}^t + \mathbf{b_y}) \tag{3.15}$$

где су са $\mathbf{W_{hx}}$, $\mathbf{W_{yh}}$ и $\mathbf{W_{hh}}$ матрице које садрже тежинске коефицијенте, а l_h и l_y неке активационе функције. Вектори b_h и b_y омогућавају учење офсета.

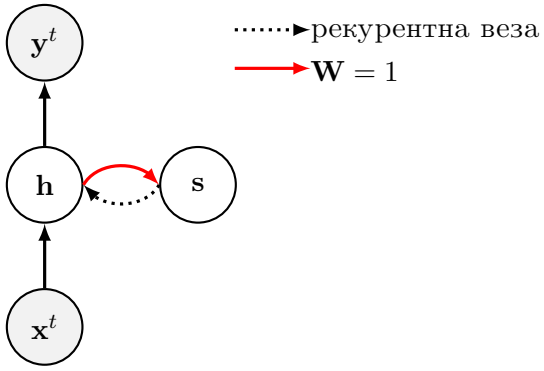
Истраживања на тему рекурентних неуралних мрежа су започела осадесетих година прошлога века. Прво је Хопфилд представио фамилију рекурентних неуралних мрежа која је имала могућности препознавања шаблона [15]. Структура мреже је иста као на слици 3.3, с тим да се улазни шаблон $\mathbf{x}$ примени само у почетном тренутку, након кога се мрежа препусти израчунавањима све док се на излазу не добије стационарна вредност. Хопфилдове мреже имају могућност да репродукује меморисане шаблоне из зашумљених шаблона које добију на улази и представљају претечу Болцманових машина и ауто-енкодера.

Рани модел за супервизовано учење секвенци је представио Џордан у [16]. Модел је *feedforward* мрежа која се састоји од једног скривеног слоја који садржи специјалне чворове. Излаз мреже је везама спојен са специјалним чворовима, који рекурентним везама, у наредном временском тренутку, достављају вредности осталим чворовима скривеног слоја. У случају да излаз мреже представља неке акције, овакав модел омогућава памећење акције из претходног временсог тренутка. Неколико модерних модела користи ову идеју. Један такав примерак је рад [17] где се на примеру преводиоца језика, при генерисању текста преведених реченица, речи са краја се враћају и користе као улаз за наредни корак. Специјални чворови могу да садрже и додатну рекурентну везу тако да за улаз имају и своје претходно стање (слика 3.4).

Елманов модел из [18] представља архитектуру која личи на мрежу са слике 3.3. За разлику од Џордановог модела, који чува тренутну вредност излаза у специјалним чворовима, Елманов модел чува тренутну вредност чворова скривеног слоја, коју у наредном временском тренутку враћа тим скривеним чворовима (слика 3.5). Елман је у свом раду тренирао мрежу алгоритмом пропагације у назад и доказао да мрежа можхе научити зависности од времена.



Слика 3.4: Шематски приказ Џорданових модела



Слика 3.5: Шематски приказ Елманових модела

3.3 Обучавање рекурентних неуралних мрежа

Док су у начелу рекурентне неуралне једноставан и моћан модел, у пракси њихово обучавање је комплексан процес. Међу главним разлозима су проблему нестајућег и експлодирајућег градијента, који су представљени у раду [19]. Критеријумска фунцкија за рекурентну неуралну мрежу представља суму губитака у појединачним временским тренуцима. Ако је дужина секвенце T, она се рачуна

$$\mathfrak{L}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{t=1}^{T} \mathfrak{L}(\hat{\mathbf{y}}, \mathbf{y}) \tag{3.16}$$

Ради оптимизације критеријумске фунцкије и даље се може користити градијенти спуст, али за рачунање градијента потребно је изменити алгоритам пропагације у назад. Модификован алгоритам се назива алгоритам пропагације у назад кроз време [20] (енг. *Backpropagation Trough Time- BPTT*). Посматрајмо једноставну рекурентну мрежу са слике 3.3, са једним скривеним рекурентним слојем. За њу важе једначине 3.14 и 3.15.

$$\mathbf{h}^t = l_h(u_t) \tag{3.17}$$
$$\mathbf{y}^t = l_y(\mathbf{o}^t) \tag{3.18}$$
$$\mathbf{o}^t = \mathbf{W_{yh}}\mathbf{h}^t + \mathbf{b_y} \tag{3.19}$$
$$\mathbf{u}^t = \mathbf{W_{hx}}\mathbf{x}^t + \mathbf{W_{hh}}\mathbf{h}^{t-1} + \mathbf{b_h} \tag{3.20}$$

Једна итерација алгоритма *BPTT* се може приказати кроз следеће поступке:

1. Иницијализују се вредности промена тежинских матрица и вектора одступања: $\Delta\mathbf{W}_{hx},\ \Delta\mathbf{W}_{hh},\ \Delta\mathbf{W}_{yh}, \Delta\mathbf{b}_h$ и $\Delta\mathbf{b}_y$ на нула векторе/матрице.
2. За сваки тренутак t од T до 1:
   1. $\Delta\mathbf{o}^t \Leftarrow l'_y(\mathbf{o}^t)\frac{\partial\mathfrak{L}(\hat{\mathbf{y}},\mathbf{y})}{\partial\hat{\mathbf{y}}^t}$
   2. $\Delta\mathbf{b}_y \Leftarrow \Delta\mathbf{b}_y + \Delta\mathbf{o}^t$
   3. $\Delta\mathbf{W}_{yh} \Leftarrow \Delta\mathbf{W}_{yh} + \Delta\mathbf{o}^t(\mathbf{h}^t)^T$
   4. $\Delta\mathbf{h}^t \Leftarrow \Delta\mathbf{h}^t + \mathbf{W}_{yh}^T\Delta\mathbf{o}^t$
   5. $\Delta\mathbf{u}^t \Leftarrow l'_h(\mathbf{u}^t)\Delta\mathbf{h}^t$
   6. $\Delta\mathbf{W}_{hx} \Leftarrow \Delta\mathbf{W}_{hx} + \Delta\mathbf{u}^t(\mathbf{v}^t)^T$
   7. $\Delta\mathbf{b}_h \Leftarrow \Delta\mathbf{b}_h + \Delta\mathbf{u}^t$
   8. $\Delta\mathbf{W}_{hh} \Leftarrow \Delta\mathbf{W}_{hh} + \Delta\mathbf{u}^t(\mathbf{h}^{t-1})^T$
   9. $\Delta\mathbf{h}^{t+1} \Leftarrow \mathbf{W}_{hh}^T\Delta\mathbf{u}^t$
3. Добијене вредности градијента искористити за алгоритам градијентног спуста.

Посматрајмо извод

$$\frac{\partial\mathfrak{L}(\hat{\mathbf{y}},\mathbf{y})}{\partial\mathbf{W}_{hh}} = \sum_{t=1}^{T}\Delta\mathbf{u}^t(\mathbf{h}^{t-1})^T \tag{3.21}$$

Вредност $\Delta\mathbf{u}^t$ се може аналитички записати

$$\Delta\mathbf{u}^t = \mathbf{W}_{yh}^T l'_y(\mathbf{o}^t)\frac{\partial\mathfrak{L}(\hat{\mathbf{y}},\mathbf{y})}{\partial\hat{\mathbf{y}}^t}\prod_{\tau=t+1}^{T}\mathbf{W}_{hh}^T l'_h(\mathbf{u}^\tau) \tag{3.22}$$

Ако су све сопствене вредности матрице $\mathbf{W}_{yh}$ мање од 1 , онда ће производ у једначини 3.22 са повећањем броја чинилаца тежити нули, и допринос суми у једначини 3.21 ће тежити нули. Другим речима грешка неће испропагирати до нижих временских тренутака, те се овај проблем назива проблем нестајућег градијента. Проблем експлодирајућег градијента настаје у случају када су сопствене вредности матрице већ од 1, али он може да се регулише избором погодних активационих функција и ограничавањем градијента помоћу регуларизације (енг. *gradient clipping*).
Једно решење проблема нестајућег градијента је скраћена пропагација у назад кроз време (енг. *Truncated Backpropagation Trough Time- TBPTT*) у коме се постави број корака у којима ће грешка пропагирати у назад [21]. Тиме се утиче на проблем нестајућег градијента, али се онемогуцује учење дужих временских зависности. Учење дужих временских зависности омогућују модерније архитектуре рекурентних неуралних мрежа.

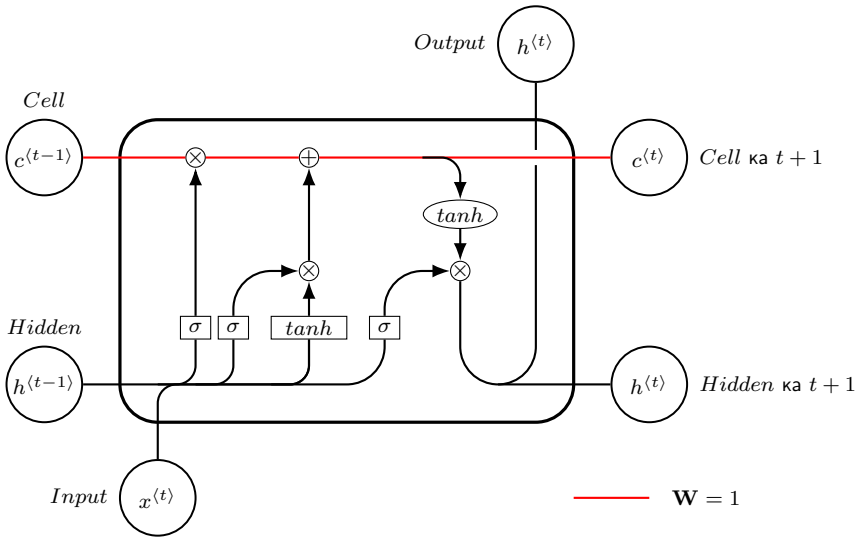
3.4 Модерне архитектуре рекурентних неуралних мрежа

Најуспешнија архитектура рекуреннтих неуралних мрежа која превазилази проблеме обучавања дужих временских зависности у секвенцама, предстаљена је у раду [22]. Овај модел се назива *Long Short-Term Memory- LSTM*. Други рад, [23],

уводи архитектуру бидирекционих рекурентних неуралних мрежа (енг. *Bidirectional Recurrent Neural Networks- BRNN*), у којој информација и из прошлости и из будућности може да се користи да се одреди излаз у сваком временском тренутку. Ово је у супротности са претходним радовима, у којима су се само улази из прошлости утицали на тренутни излаз. Такође, *LSTM* и *BRNN* је могуће комбиновати да би се добила *Bidirectional LSTM* архитектура.

3.4.1 LSTM модел

Hochreiter и *Schmidhuber* у [22] су увели *LSTM* модел примарно како би превазишли проблем нестајућег граддијента. Овај модел представља стандардну рекурентну нуралну мрежу са скривеним слојем (слика 3.3), али класични чвор је замењен са меморијском ћелијом (слика 3.6).



Слика 3.6: *LSTM* меморијска ћелија

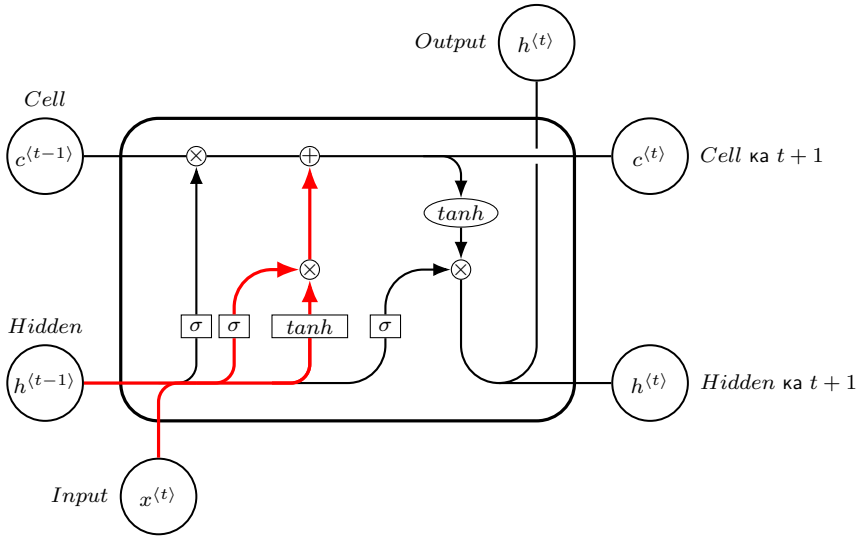
Кључна ствар у *LSTM* чвору је стање ћелије (енг. *cell state*) преко ког је омогућено кретање информације без промене. То се остварује тако што су тежински коефицијенти рекурентних веза између ћелијских стања фиксирани на 1. *LSTM* има могућност да дода или одузме информације из стања ћелије преко структуре која се назива *gate*. *Gate* се састоји од сигмоид активационе функције, која се примењује на један вектор улаза. Сигмоид функција има вредности између 0 и 1 и скалраним множењем са другим улазним вектором се одлуцјује колико се информације тог другог вектора преноси даље. *LSTM* чвор садржи три *gate*-а.

- **Input gate**
  Овај *gate* израчунава кандидата за ново стање ћелије. Састоји се од два дела (слика 3.7). Први део рачуна новог кандидата којим ће се ажуририати вредност стања ћелије $\tilde{C}_t$, а други део i_t одлучује који део новог кандидата ће се додати на тренутно стање ћелије.

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{3.23}$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \tag{3.24}$$

где је са $[\mathbf{h}_{t-1}, \mathbf{x}_t]$ означен здружени вектор претходног излаза ћелије и тренутног вектора улаза.

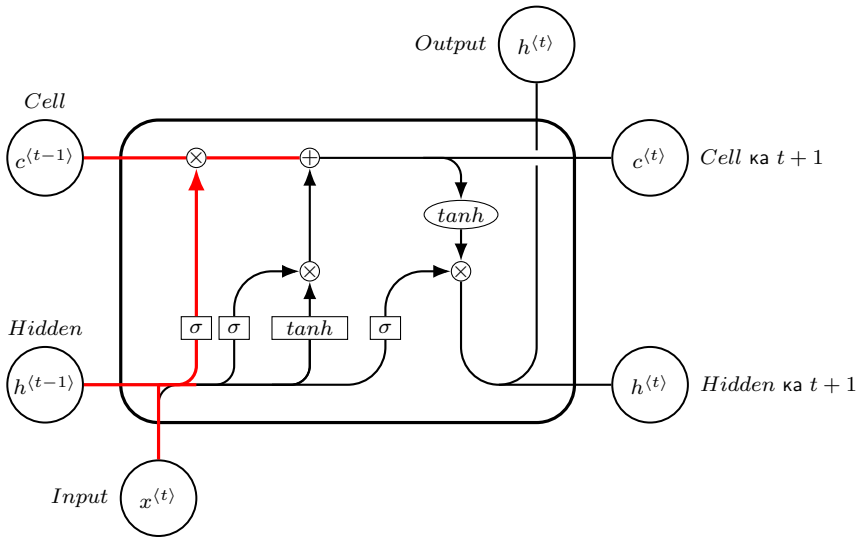


Слика 3.7: *Input gate*

- **Forget gate**
  Овај *gate* одлучује колико ће се информације из претходног ћелијског стања $\mathbf{C}_{t-1}$ задржати. Треба напоменути да је овај *gate* додат накнадно у раду [24]. Одлука се доноси на основу излаза сигмоид функције који посматра здружени вектор тренутног улаза и претходног излаза.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{3.25}$$

Ново стање ћелије се добија комбинацијом *forget gate*-а са претходним ста-



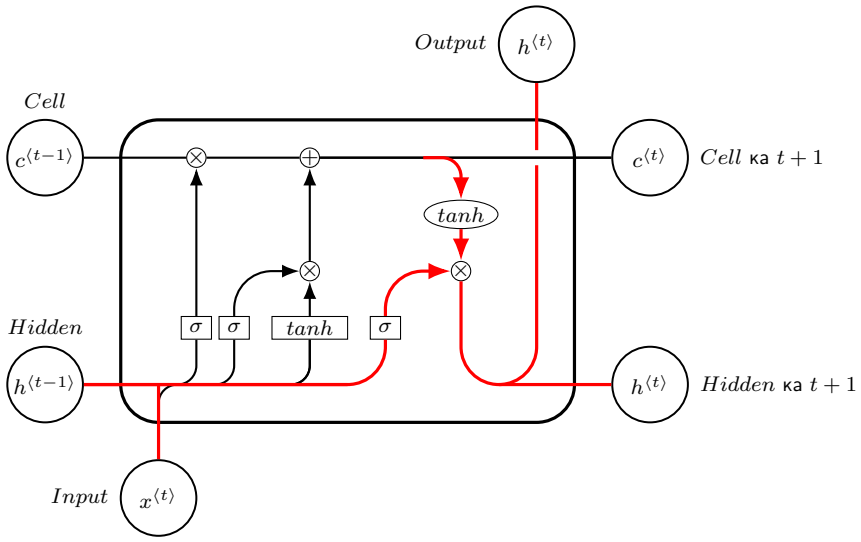
Слика 3.8: *Forget gate*

њем ћелије и комбинацијом *input gate*-а са кандидатом за ново стање

$$\mathbf{C}_t = \mathbf{f}_t \cdot \mathbf{C}_{t-1} + \mathbf{i}_t \cdot \tilde{\mathbf{C}}_{t-1} \tag{3.26}$$

- **Output gate**
  Излаз се рачуна на основу стања ћелије које се прво пропусти кроз tanh, како би се вредности ограничиле на интервал $[-1,\ 1]$, а затим се тај вектор скаларно помножи са вектором који одлучује колико ће се стања ћелије проследити на излаз

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{3.27}$$
$$\mathbf{h}_t = \tanh(\mathbf{C}_t) \tag{3.28}$$



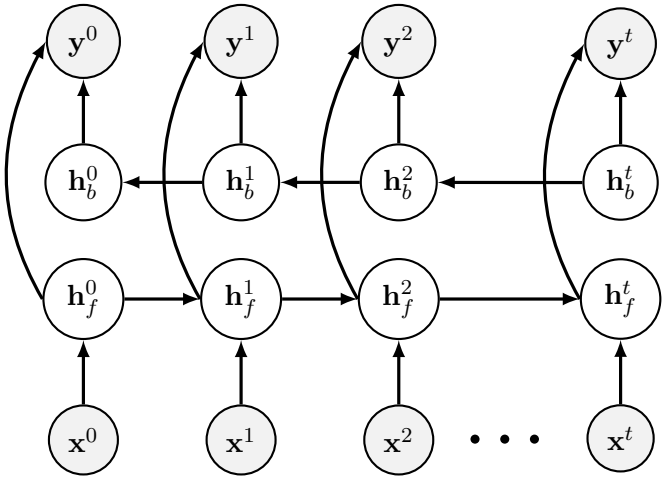
Слика 3.9: *Output gate*

3.4.2 BRNN модел

У овој архитектури, постоје два слоја чворова у скривеном слоју , и оба слоја су повезана са улазним и излазним слојем. Један слој има рекурентне везе које носе информацију ис прошлих временских тренутака, док други носи информацију из наредних временских тренутака. За тренутак t и улазни вектор $\mathbf{x}_t$, стања и излаз мреже се могу рачунати на следећи начин

$$\overrightarrow{\mathbf{h}}_t = \sigma\left(\mathbf{W}^f_{hx}\mathbf{x}_t + \mathbf{W}^f_{hh}\overrightarrow{\mathbf{h}}_{t-1} + \mathbf{b}^f_h\right) \tag{3.29}$$
$$\overleftarrow{\mathbf{h}}_t = \sigma\left(\mathbf{W}^b_{hx}\mathbf{x}_t + \mathbf{W}^b_{hh}\overleftarrow{\mathbf{h}}_{t-1} + \mathbf{b}^b_h\right) \tag{3.30}$$
$$\hat{\mathbf{y}}_t = \sigma\left(\mathbf{W}^f_{yh}\overrightarrow{\mathbf{h}}_t + \mathbf{W}^b_{yh}\overleftarrow{\mathbf{h}}_t + \mathbf{b}_y\right) \tag{3.31}$$

Са датом улазном секвенцом, прво се израчунају стања *forward* слоја $\overrightarrow{\mathbf{h}}_t$ почевши од првог одбирка, а затим се израчунају стања *backward* слоја $\overleftarrow{\mathbf{h}}_t$ почевши од последњег одбирка секвенце. Након тога се, уз по једначини 3.31 израчуна вектор излаза. За обучавање, може се користити алгоритам *BPTT*. Једна од мана *BRNN* модела је што се не може имплементирати у реалном времену, јер је немогуће сазнати вредност улазних одбирака у будућности.



Слика 3.10: Шематски приказ одмотане бидирекционе рекурентне неуралне мреже у времену

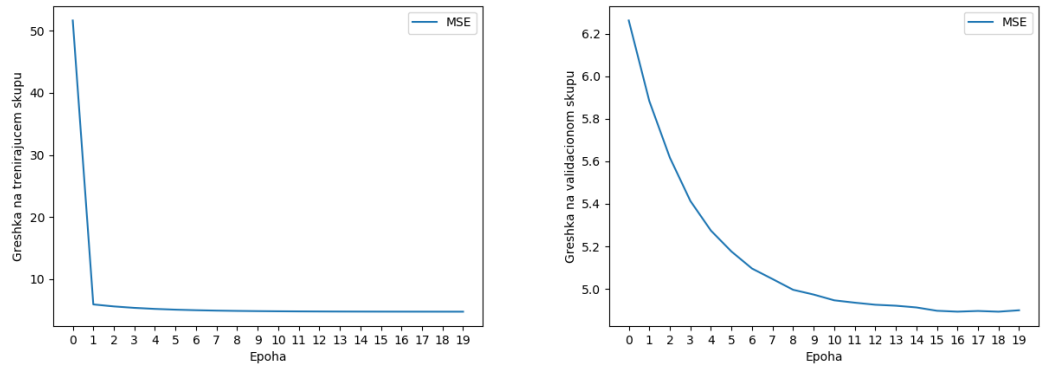
Бидирекционе мреже могу да се комбинују са $LSTM$ моделом тако што се уместо обичних RNN чворова користе $LSTM$ ћелије.

4 | Резултати обучавања модела

У овом поглављу ће бити представљени разултати обучавања различитих модлеа рекурентних неуралних мрежа.
Обележја говора и обележја видеа се користе као улазне и излазне секвенце предложених рекурентних неуралних мрежа. Обележја су добијена из 17 h дугог видео материјала доступног на сајту *YouTube*. Видеи представљају недељна обраћања нацији председника Обаме, у периоду од 2009. до 2016. године. Видеи су морали бити у мањој резолуцији, 480*p*, како би издвајање облика уста било урађено у разумном периоду.
У поглављу 1 је објашњен начин на који се долази до улазне секвенце. Може се приметити да је периода између два вектора улазне секвенце, већа од периоде између два вектора секвенце излаза. Периода између два МФКК је условљена дужином прозора на које се деле аудио сигнали, као и дужином преклапања два суседна прозора, док је периода између два облика уста условљена бројем фрејмова у секудни видеа. Стога је потребно извршити интерполацију излазних секвенци, како би се улазне и излазне секвенце свеле на исту периоду. Поред тога, обележја говора се нормализују. Ради потреба обучавања, целокупан сет се дели на тренирајући и валидациони скуп са вероватноћом 0.2.
Сви модели су реализовани уз помоћ библиотеке *TensorFlow* [33], где је коришћен *Keras API*. Преложени су различити *LSTM* и *BRNN* модели, за чије обучавање је изабран *Adam* [14] оптимизатор. За вредност брзине обучавања α узета је вредност 0.01. Ради спречавања проблема експлодирајућег градијента, за градијент за све параметре градијент се клипује на вредност 100. За критеријумску функцију је узета средња квадратна грешка (*MSE*). Хардверске спецификације рачунара на коме је извршено обучавање модела су: процесор *Intel i*5-8300*H* са 16 GB *RAM* меморије и графичком картицом *NVIDIA GeForce* 1050*Ti* са 4 GB меморије. Време тренирања модела зависи од комплесности модела, али у просеку за 20 епоха је потребно неколико сати.

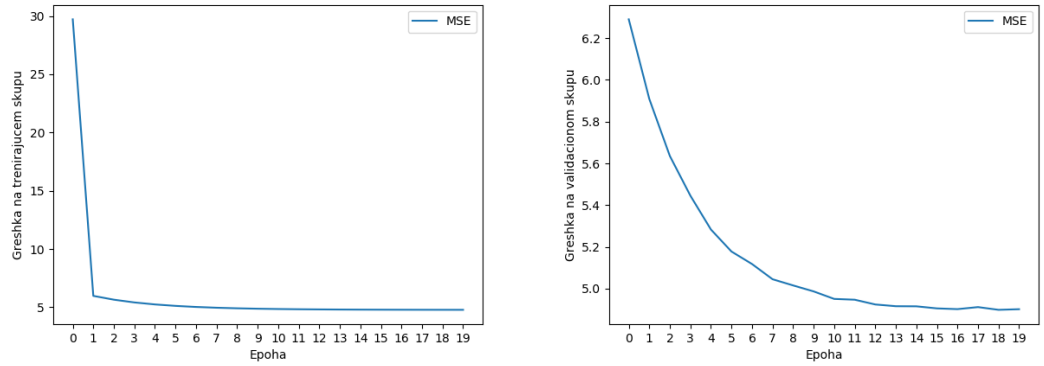
4.1 LSTM модели

У овом делу биће приказани разултати обучавања *LSTM* модела. Параметар који варира је димензија стања ћелије n у интервалу од 20 до 150. Првобитно је покушано са једним слојем скривених рекуретних *LSTM* ћелија, а касније и са вишеслојним мрежама. Генерално, овакви модели су засењени перформансом *BRNN*. Додавање слојева није показало значајније побољшање, али време потребно за тренирање је значајно продужено.



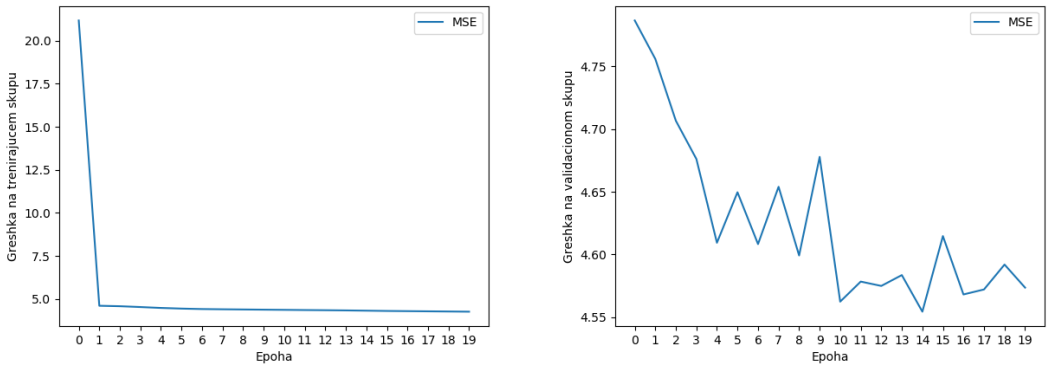
(а) MSE на тренирајућем скупу (б) MSE на валидационом скупу

Слика 4.1: Једнослојна $LSTM$ мрежа са $n = 30$



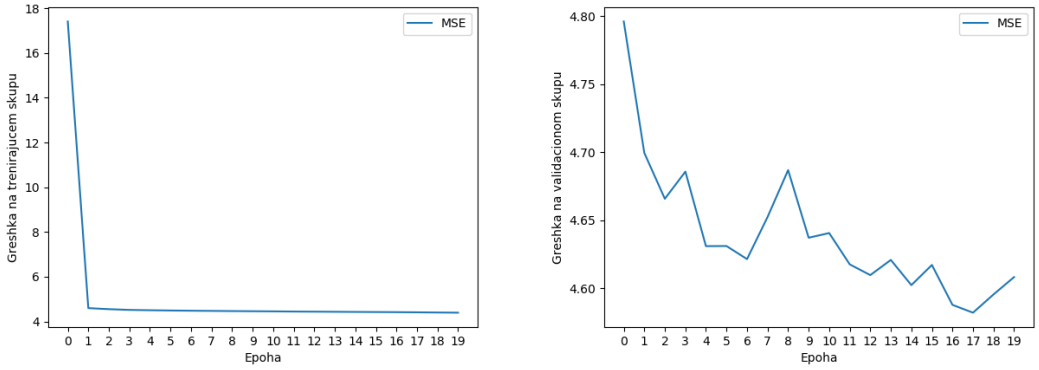
(а) MSE на тренирајућем скупу (б) MSE на валидационом скупу

Слика 4.2: Једнослојна $LSTM$ мрежа са $n = 60$



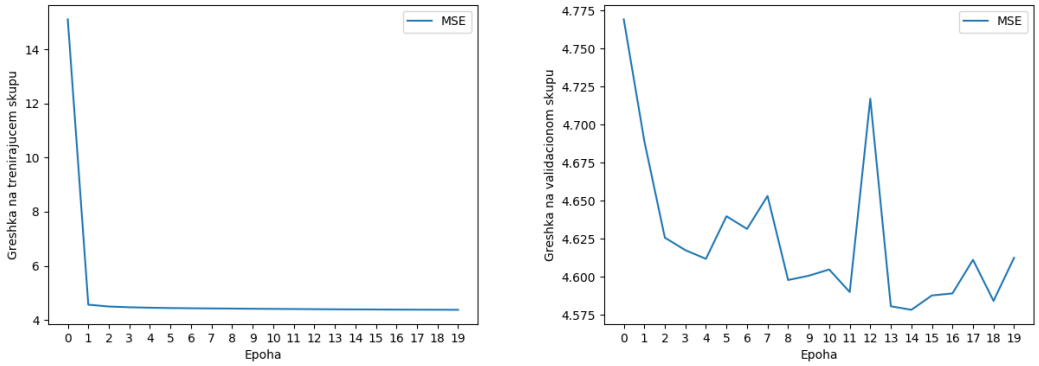
(а) MSE на тренирајућем скупу (б) MSE на валидационом скупу

Слика 4.3: Једнослојна $LSTM$ мрежа са $n = 90$



(а) MSE на тренирајућем скупу (б) MSE на валидационом скупу

Слика 4.4: Једнослојна $LSTM$ мрежа са $n = 150$

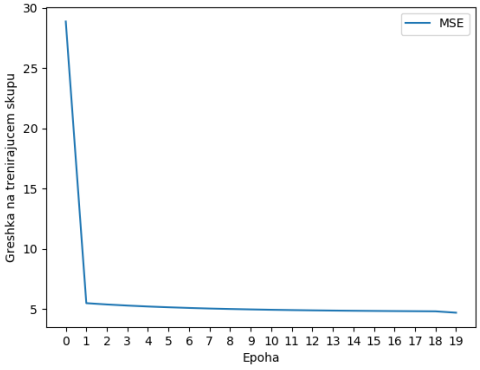


(а) MSE на тренирајућем скупу (б) MSE на валидационом скупу

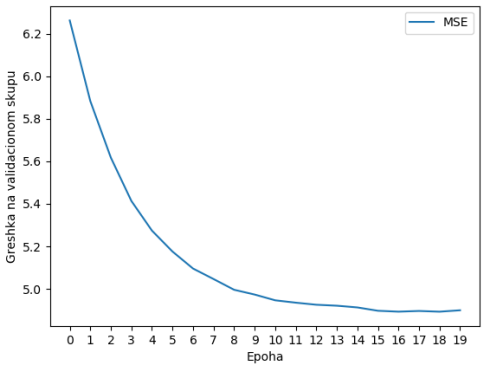
Слика 4.5: Једнослојна $LSTM$ мрежа са $n = 150$

4.2 BRNN модели

У овом делу ће бити приказани резултати $BRNN$ модела, који за рекурентне чворове имају $LSTM$ ћелије. Може се приметити да ове мреже имају значајно бољи учинак од обичних $LSTM$ мрежа. То се може оправдати запажањем да човек док говори, унапред зна која су наредни фонети у оквиру једне речи и тако будуци фонеми утичу на тренутни положај уста.

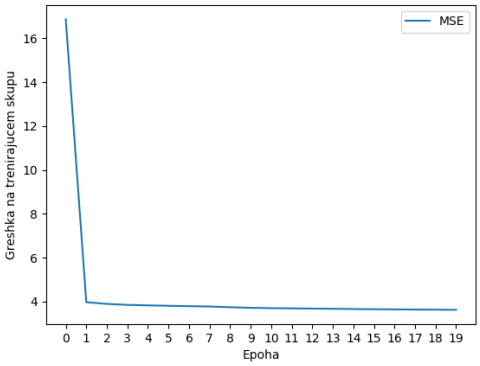


(а) MSE на тренирајућем скупу

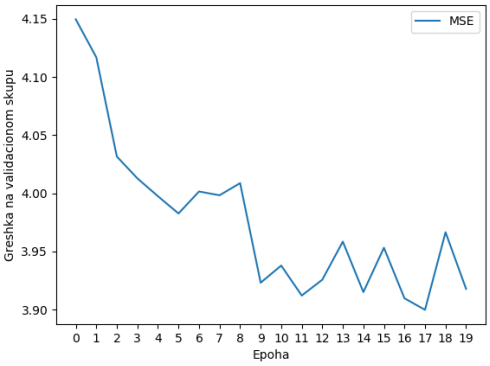


(б) MSE на валидационом скупу

Слика 4.6: Једнослојна $BRNN$ мрежа са $n = 30$

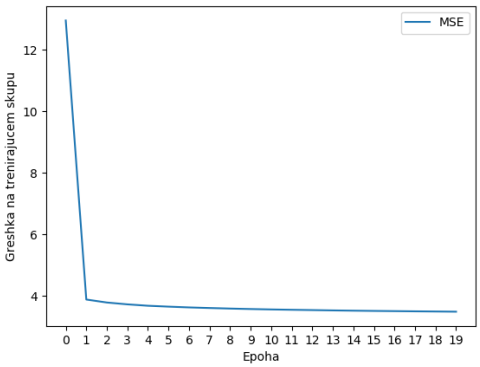


(а) MSE на тренирајућем скупу

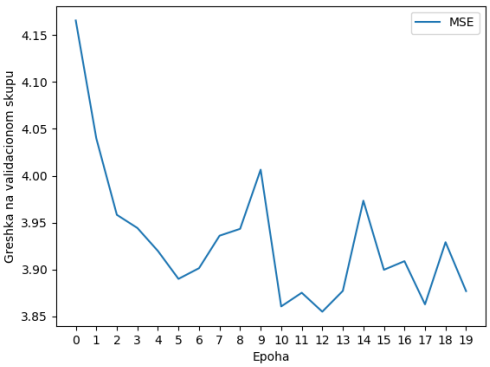


(б) MSE на валидационом скупу

Слика 4.7: Једнослојна $BRNN$ мрежа са $n = 60$

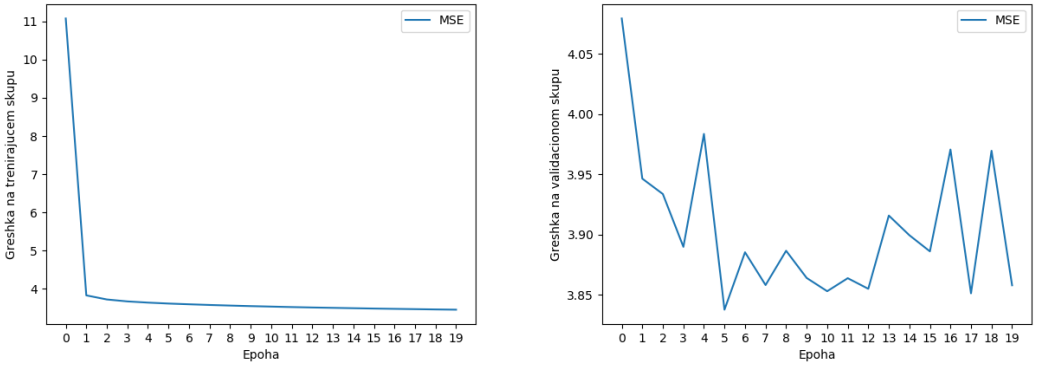


(а) MSE на тренирајућем скупу



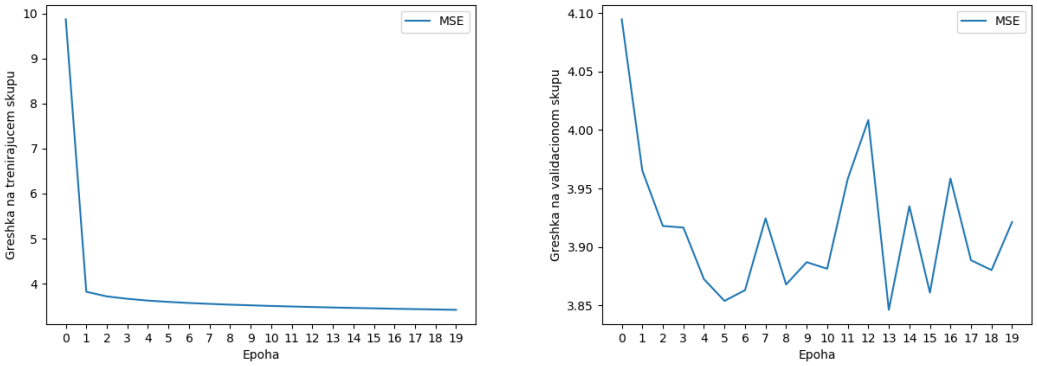
(б) MSE на валидационом скупу

Слика 4.8: Једнослојна $BRNN$ мрежа са $n = 90$



(а) MSE на тренирајућем скупу (б) MSE на валидационом скупу

Слика 4.9: Једнослојна $BRNN$ мрежа са $n = 150$



(а) MSE на тренирајућем скупу (б) MSE на валидационом скупу

Слика 4.10: Једнослојна $BRNN$ мрежа са $n = 150$

5 | Синтеза видеа

6 | Закључак

Литература

[1] Ђуровић, Ж. Материјали са предмета Обрада и препознавање говора

[2] Ђуровић, Ж. Материјали са предмета Препознавање облика

[3] Тадић, П. Материјали са предмета Машинско учење

[4] Делић, В. Анализа мел-фреквенцијских кепстралних коефицијената као обележја коришћених при аутоматском препознавању говорника

[5] Christoph Bregler, Michele Covell, Malcolm Slaney. Video Rewrite: Driving Visual Speech with Audio. Interval Researh Corporation, 1997.

[6] Supasorn Suwajanakorn, Steven M. Seitz, Ira Kemelmacher-Shlizerman. Synthesizing Obama: Learning Lip Sync from Audio. University of Washington, 2017.

[7] http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[8] https://ffmpeg.org/

[9] https://github.com/slhck/ffmpeg-normalize

[10] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[11] Matthew D. Zeiler. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701, 2012.

[12] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. The Journal of Machine Learning Research, 12:2121–2159, 2011.

[13] Tijmen Tieleman and Geoffrey E. Hinton. Lecture 6.5- RMSprop: Divide the gradient by a running average of its recent magnitude, 2012.

[14] Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization, 2014.

[15] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 79(8):2554–2558, 1982.

[16] Michael I. Jordan. Serial order: A parallel distributed processing approach. Technical Report 8604, Institute for Cognitive Science, University of California, San Diego, 1986.

[17] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems, pages 3104–3112, 2014.

[18] Jeffrey L. Elman. Finding structure in time. Cognitive science, 14(2):179–211,1990.

[19] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks,5(2), 157–166

[20] Werbos, P. (1990). Backpropagation through time: what it does and how to do it. Proceedings of the IEEE, 78(10):1550–1560.

[21] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. Neural Computation, 1(2):270–280,1989.

[22] Sepp Hochreiter and J¨urgen Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.

[23] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on, 45(11):2673–2681, 1997.

[24] Felix A. Gers, J¨urgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. Neural computation, 12(10):2451–2471, 2000.

[25] https://opencv.org/

[26] Paul Viola, Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. Conference on Computer Vision and Pattern Recognition, 2001.

[27] Shaoqing Ren, Xudong Cao, Yichen Wei, Jian Sun. Face Alignment at 3000 FPS via Regressing Binary Features, 2014.

[28] Blanz, V. and Vetter, T. (1999). A Morphable Model for the synthesis of 3D faces. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pages 187–194. ACM Press/Addison-Wesley Publishing Co.

[29] Cootes, T., Edwards, G., and Taylor, C. (2001). Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):681–685.

[30] Patrik Huber, Guosheng Hu, Rafael Tena, Pouria Mortazavian, Willem P. Koppen, William Christmas, Matthias Rätsch and Josef Kittler. A Multiresolution 3D Morphable Face Model and Fitting Framework. Centre for Vision, Speech & Signal Processing, University of Surrey. 2016

[31] https://github.com/patrikhuber/eos

[32] https://en.wikipedia.org/wiki/Wavefront_.obj_file

[33] https://www.tensorflow.org/

Списак слика

Списак табела

Списак скраћеница

3DMM 3D Morphable Model. 15, 16

AAM Active Appearance Model. 15

BPTT Backpropagation Trough Time. 23, 24, 27

BRNN Bidirectional Recurrent Neural Networks. 25, 27, 29, 31–33, 39

LBF Local Binary Features. 13, 14

LBP Linear Binary Pattern. 11, 13

LSTM Long Short-Term Memory. 24, 25, 28–31, 39

MSE Mean Square Error. 29–33, 39

PCA Principal Component Analysis. 15, 16

RNN Recurrent Neural Networks. 28

SFM Surrey Face Model. 16

TBPTT Truncated Backpropagation Trough Time. 24

ДФТ дискретна Фуријеова трансформација. 6, 8

МФКК мел-фреквенцијски кепстрални коефицијенти. 3, 6, 8–10, 29, 38