



# УНИВЕРЗИТЕТ У НОВОМ САДУ ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА  
НОВИ САД

Департман за рачунарство и аутоматику

Одсек за рачунарску технику и рачунарске комуникације

## ЗАВРШНИ (BACHELOR) РАД

Кандидат: Алекса Арсић

Број индекса: РА 119-2015

Тема рада: Једно решење система за праћење пажње возача уз помоћ обраде слика моделима машинског учења

Ментор рада: доц. др Богдан Павковић

Нови Сад, август, 2020.



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА  
21000 НОВИ САД, Трг Доситеја Обрадовића 6

## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, <b>РБР:</b>	
Идентификациони број, <b>ИБР:</b>	
Тип документације, <b>ТД:</b>	Монографска документација
Тип записа, <b>ТЗ:</b>	Текстуални штампани материјал
Врста рада, <b>ВР:</b>	Завршни (Bachelor) рад
Аутор, <b>АУ:</b>	<b>Алекса Арсић</b>
Ментор, <b>МН:</b>	<b>доц. др Богдан Павковић</b>
Наслов рада, <b>НР:</b>	<b>Једно решење система за праћење пажње возача уз помоћ обраде слика моделима машинског учења</b>
Језик публикације, <b>ЈП:</b>	Српски / латиница
Језик извода, <b>ЈИ:</b>	Српски
Земља публикација, <b>ЗП:</b>	Република Србија
Уже географско подручје, <b>УГП:</b>	Војводина
Година, <b>ГО:</b>	<b>2020.</b>
Издавач, <b>ИЗ:</b>	Ауторски репринт
Место и адреса, <b>МА:</b>	Нови Сад; трг Доситеја Обрадовића 6
Физички опис рада, <b>ФО:</b> (поглавља/страница/ цитата/табела/слика/графика/прилога)	<b>7/78/25/7/21/18/0</b>
Научна област, <b>НО:</b>	Електротехника и рачунарство
Научна дисциплина, <b>НД:</b>	Рачунарска техника
Предметна одредница/Кључне речи, <b>ПО:</b>	<b>Машинско учење, дубоко учење, вештачке неуронске мреже, конволутивне неуронске мреже, обрада слике, пажња возача</b>
<b>УДК</b>	
Чува се, <b>ЧУ:</b>	У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, <b>ВН:</b>	
Извод, <b>ИЗ:</b>	Систем за праћење пажње возача заснован на обради слике уз помоћ принципа машинског учења и конволутивним неуронских мрежа. У раду је приказана теоријска позадина, решења за генерисање и означавање скупова података, апликација за праћење пажње возача, као и евалуација решења.
Датум прихватања теме, <b>ДП:</b>	
Датум одбране, <b>ДО:</b>	
Чланови комисије, <b>КО:</b>	Председник: доц. др Јелена Ковачевић
	Члан: проф. др Драган Самарџија
	Члан, ментор: доц. др Богдан Павковић
	Потпис ментора

Accession number, <b>ANO</b> :		
Identification number, <b>INO</b> :		
Document type, <b>DT</b> :		Monographic publication
Type of record, <b>TR</b> :		Textual printed material
Contents code, <b>CC</b> :		Bachelor Thesis
Author, <b>AU</b> :		Aleksa Arsić
Mentor, <b>MN</b> :		Bogdan Pavković, PhD
Title, <b>TI</b> :		One solution of driver monitoring system using image processing with machine learning models
Language of text, <b>LT</b> :		Serbian
Language of abstract, <b>LA</b> :		Serbian
Country of publication, <b>CP</b> :		Republic of Serbia
Locality of publication, <b>LP</b> :		Vojvodina
Publication year, <b>PY</b> :		2020.
Publisher, <b>PB</b> :		Author's reprint
Publication place, <b>PP</b> :		Novi Sad, Dositeja Obradovica sq. 6
Physical description, <b>PD</b> : (chapters/pages/ref./tables/pictures/graphs/appendixes)		7/78/25/7/21/18/0
Scientific field, <b>SF</b> :		Electrical Engineering
Scientific discipline, <b>SD</b> :		Computer Engineering, Engineering of Computer Based Systems
Subject/Key words, <b>S/KW</b> :		Machine learning, deep learning, artificial neural networks, convolutional neural networks, image processing, driver attention
<b>UC</b>		
Holding data, <b>HD</b> :		The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, <b>N</b> :		
Abstract, <b>AB</b> :		Driver monitoring system based on image processing with machine learning principals and convolutional neural networks. In this paper, theoretical background is shown, solutions for generating and labeling datasets, driver monitoring application and testing methods and results.
Accepted by the Scientific Board on, <b>ASB</b> :		
Defended on, <b>DE</b> :		
Defended Board, <b>DB</b> :	President:	Jelena Kovačević, PhD
	Member:	Dragan Samardžija, PhD
	Member, Mentor:	Bogdan Pavković, PhD
		Menthor's sign

---

## Захвалност

Велику захвалност дугујем академском ментору др Богдану Павковићу и изузетном стручњаку у пољу машинског учења др Велибору Илићу који су ми помогли приликом израде овог рада.

Највише се захваљујем породици, пријатељима и свим драгим људима који су ми пружили несебичну подршку током школовања, али и током израде овог рада. Хвала вам!

## Садржај

Захвалност .....	3
Садржај .....	4
Списак слика .....	6
Списак табела.....	7
Списак графикана.....	8
Списак формула.....	9
1    Увод .....	10
2    Теоријске основе .....	13
2.1    Машинско учење .....	13
2.1.1    Надгледано учење .....	14
2.1.2    Ненадгледано учење .....	15
2.1.3    Полунадгледано учење .....	15
2.2    Дубоко учење .....	15
2.2.1    Вештачке неуронске мреже.....	16
2.2.2    Примене дубоког учења .....	18
2.3    Конволутивне неуронске мреже .....	20
2.3.1    Улаз у мрежу .....	22
2.3.2    Конволутивни слој .....	23
2.3.3    Сажимајући слој .....	25

2.3.4	Потпуно повезани слој .....	26
2.3.5	Обучавање, параметри модела и хипер-параметри .....	27
3	Концепт решења .....	31
3.1	Генерисање и означавање скупова података .....	32
3.1.1	Генерисање скупова података .....	32
3.1.2	Означавање скупова података .....	33
3.1.2.1	Режим рада откривања лица .....	35
3.1.2.2	Режим рада откривања елемената лица .....	36
3.1.2.3	Режим рада откривања контура ока .....	37
3.1.2.4	Излазне датотеке апликације за означавање .....	38
3.2	Апликација за праћење пажње возача .....	42
3.2.1	Модел конволутивних неуронских мрежа .....	42
3.2.2	Процес обучавања мрежа .....	44
3.2.3	Структура апликације .....	45
4	Програмско решење .....	48
4.1	Апликација за означавање података .....	48
4.2	Апликација за праћење пажње возача .....	53
4.3	Концепт модела фокусиране пажње .....	56
5	Евалуација .....	58
5.1	Евалуација апликације за означавање података .....	58
5.2	Евалуација апликације за праћење пажње возача .....	60
5.2.1	Евалуација првог модела неуронске мреже .....	61
5.2.2	Евалуација другог модела неуронске мреже .....	64
5.2.3	Евалуација трећег модела неуронске мреже .....	68
5.2.4	Могућност искључивања другог модела .....	71
5.2.5	Временска потрошња појединих целина програмског кода .....	72
6	Закључак .....	74
7	Референце .....	76

## Списак слика

Слика 2.1 Структура дубоке неуронске мреже [9].....	17
Слика 2.2 Структура поља вештачке интелигенције према [7], [10].....	18
Слика 2.3 <i>deepart.io</i> у комбинацији са Ван Гоговом Звезданом ноћи.....	20
Слика 2.4 Пример архитектуре конволутивне неуронске мреже [14].....	21
Слика 2.5 Преглед РГБ канала .....	22
Слика 2.6 Приказ операције конволуције [15].....	23
Слика 2.7 Пример <i>Max-pool</i> и <i>Average-pool</i> метода са кораком = 2.....	26
Слика 2.8 Потпуно повезани слој вештачких неуронских мрежа .....	27
Слика 2.9 Нежељена и жељена стања модела неуронских мрежа [17].....	29
Слика 3.1 Илустровани приказ тачке 1, 2 и 3, генерисање скупова података.....	32
Слика 3.2 Структура апликације за означавање података.....	34
Слика 3.3 Режим рада откривања лица .....	36
Слика 3.4 Режим рада откривања елемената лица .....	37
Слика 3.5 Режим рада откривања контура ока .....	38
Слика 3.6 Архитектура конволутивне неуронске мреже [20].....	42
Слика 3.7 Процес обучавања неуронске мреже .....	44
Слика 3.8 Структура апликације за праћење пажње возача.....	45
Слика 3.9 Финална апликација за праћење пажње возача .....	47
Слика 4.1 Апликација за означавање података .....	49
Слика 4.2 Апликација за праћење пажње возача .....	53
Слика 4.3 Концепт модела фокусиране пажње .....	57

## Списак табела

Табела 3.1 Резолуције генерисаних скупова података .....	33
Табела 3.2 Димензије и параметри слојева првог *(другог) **(трећег) модела кориштене архитектуре неуронских мрежа .....	43
Табела 5.1 Проценат тачних предикција над тест сетом првог модела .....	63
Табела 5.2 Проценат тачних предикција над тест сетом другог модела.....	67
Табела 5.3 Проценат тачних предикција над тест сетом трећег модела.....	70
Табела 5.4 Временска потрошња делова кода који се извршавају на централној процесорској јединици .....	73
Табела 5.5 Временска потрошња делова кода који се извршавају на графичком процесору.....	73



## Списак графикана

Графикон 2.1 Грешка класификације на основу <i>ImageNet</i> скупа података према [11]	19
Графикон 5.1 Просечно време читања .csv датотеке свих режима рада	59
Графикон 5.2 Време уписа у .csv датотеке	59
Графикон 5.3 Време уписа у нормализоване .csv датотеке	60
Графикон 5.4 Тачност обучавања и валидације првог модела	61
Графикон 5.5 Губици првог модела	61
Графикон 5.6 $x$ координата центра лица	62
Графикон 5.7 Ширина лица	62
Графикон 5.8 Тачност обучавања и валидације другог модела	64
Графикон 5.9 Губици другог модела	64
Графикон 5.10 $x$ координата носа	65
Графикон 5.11 Модел 2 - поглед усмерен лево	66
Графикон 5.12 Тачност обучавања и валидације трећег модела	68
Графикон 5.13 Губици трећег модела	68
Графикон 5.14 $y$ координата центра зенице	69
Графикон 5.15 $x$ координата централне горње тачке ока	69
Графикон 5.16 Предикције првог и другог модела лево око - $x$	71
Графикон 5.17 Предикције првог и другог модела лево око - $y$	72

## Списак формула

(2.1) Процес конволуције.....	24
(2.2) РЕЛУ активацијска функција .....	24
(2.3) Излазне димензије конволутивног слоја .....	24
(2.4) Број параметара конволутивног слоја .....	25
(2.5) Број тежинских коефицијената потпуно повезаног слоја.....	27
(3.1) Нормализација излазних вредности апликације за означавање.....	39
(4.1) <i>Simple moving average</i> .....	55

## 1 Увод

Аутомобилска индустрија последњих се година нашла пред многим захтевима и препрекама на које треба дати одговор, између чега, највећи притисак на њу врше дигитализација и поштравање сигурносних захтева, али и тежња да се обезбеди што боља сигурност возила. Интеграција хардвера и програмских решења знатно је подигла ниво функционалности и ниво комплексности аутомобила што је резултирало све већом потребом за озбиљним стручним кадром и подизањем цена, али и оставило траг на задовољство купаца. [1] Аутомобилска индустрија је сваког дана све ближе тачки у којој ће аутономни аутомобили бити свакодневница и као такви већ сада користе све већи број уграђених система који се ослањају на обраду фотографије. Број таквих система у сталном је порасту, а најчешће се могу видети као системи за асистирање приликом паркирања или чак у механизмима који покушавају да спрече колизију возила са другим објектима. [2]

Да би се обезбедила функционалност система који се ослањају на обраду фотографије потребно је детектовати, класификовати и локализовати објекте који се налазе у окружењу возила. Ово се може постићи техникама машинског учења (енг. *Machine learning*) које представља групу специфичних алгоритама који рачунарима омогућавају да на основу стеченог искуства доносе одлуке без да су експлицитно програмирани за то. Осим што се ови алгоритми могу користити за потенцијалне проблеме и њихова решења ван возила, они се могу искористити и за праћење нивоа пажње возача. Прецизније, могу се користити технике које спадају у подскуп техника

машинског учења, а ослањају се на коришћење вештачких неуронских мрежа. Оне се још називају и технике дубоког учења (енг. *Deep learning*). [3], [4]

Вештачке неуронске мреже чине скупови вештачких неурона који су повезани конекцијама које прослеђују сигнале до других вештачких неурона, а те конекције још се називају и синапсе. Вештачки неурон који прими сигнал га обрађује и прослеђује другим вештачким неуронима са којима је повезан, а излазни сигнал који се прослеђује израчунава се уз помоћ нелинеарне математичке функције. Обучавањем вештачких неуронских мрежа формира се математички модел који се води искуством и ако је добро формиран може да предвиди шта би требало да се појави на излазу неуронске мреже без да је такав случај видео приликом обучавања. Само обучавање врши се уз помоћ сета података (енг. *Dataset*) који неуронској мрежи омогућава да формира потребан математички модел. Архитектура вештачких неуронских мрежа постоји мноштво, међутим за обраду фотографије користе се конволутивне неуронске мреже које чине основу овог рада.

Овај рад описује једно решење система за праћење пажње возача применом скупа конволутивних неуронских мрежа за одређивање стања возача које би потенцијално угрозило његову сигурност, али и осталих учесника у саобраћају. Скуп конволутивних неуронских мрежа чине три мреже од којих свака имплементира различите функционалности. Прва мрежа проналази лице (ако постоји) и центар оба ока. Друга мрежа, на фотографији лица, проналази центар оба ока, нос и горњу и доњу усну, док трећа мрежа, на фотографији ока, проналази тачке од интереса и одређује да ли је око отворено или затворено. Након проласка сигнала кроз ове три неуронске мреже процењује се да ли се возач налази у стању ниске пажње. Одређивањем таквог стања ниске пажње, имплементирана је могућност благовременог давања упозорења, у виду звучног сигнала, како би се избегле потенцијално опасне или чак погубне ситуације.

Осим поменутог решења, имплементирани су и алати за што лакше генерисање и манипулацију скупом података, који се користио за обучавање целокупног система. Имплементација ових алата укључује: алат за генерисање скупова фотографија које су кориштене за обучавање система, алат за означавање скупова података кроз све фазе обучавања, алата за проширивање скупова података (енг. *Data set augmentation*) и разних помоћних алата за манипулацију тих скупова података.

Полазна тачка за реализацију овог рада представљала је упознавање са методама и техникама дубоког и машинског учења, упознавање са принципима конволутивних неуронских мрежа, али и савладавање коришћења есенцијалних програмских алата и софтвера. Сам систем за праћење пажње возача реализован је уз помоћ *Tensorflow 2.0 Python* библиотеке која у себи садржи *Keras* модул потребан за дефинисање модела конволутивних неуронских мрежа. Остали додатни алати реализовани су уз помоћ *Python* или *C#* програмских језика.

У наставку је дат кратак опис структуре дипломског рада:

- 1. Теоријске основе** – у овом поглављу дат је кратак преглед теоријских основа машинског учења, дубоког учења и конволутивних неуронских мрежа. Описане су специфичности у различитим типовима приступа учењу у алгоритмима дубоког учења, као и практичне примене дубоког учења у разним сферама. Акценат је дат на конволутивне неуронске мреже јер оне представљају основу имплементације решења овог проблема. Такође, детаљније је описана њихова структура.
- 2. Концепт решења** – у овом поглављу описани су концепти програмских решења генерисања и означавања скупова података потребних за обучавање неуронских мрежа и излазне датотеке из програма за означавање. Дат је опис архитектуре конволутивних неуронских мрежа и структура саме апликације за праћење пажње возача.
- 3. Програмско решење** – у овом поглављу дат је детаљан опис битних структура, класа и функција програмских решења апликације за означавање података и финалне апликације, те је дат осврт на систем са становишта модела фокусиране пажње.
- 4. Евалуација** – у овом поглављу разматрају се резултати са становишта временске потрошње и меморијске потрошње апликације за означавање скупова података. Анализирани су модели са становишта тачности над очекиваним резултатима над тест подацима, али и могућност одбацивања дела решења.
- 5. Закључак** - у овом поглављу дат је критички осврт на целокупан рад и постигнуте резултате система за праћење пажње возача.

## 2 Теоријске основе

Ово поглавље покрива кратке теоријске основе машинског учења, дубоког учења и конволутивних неуронских мрежа које представљају основу овог рада.

### 2.1 Машинско учење

Машинско учење представља грану проучавања специфичне фамилије алгоритама која поседује могућност аутоматског побољшања излазних резултата на основу стеченог искуства. Често се за машинско учење каже да оно представља подскуп вештачке интелигенције из разлога јер алгоритми машинског учења проналазе узорке над великом количином улазних података, креирају математички модел и на основу таквог стеченог искуства могу да ураде предикцију или направе одлуку без да су експлицитно програмирани за то.

Другим речима, машинско учење јесте поступак у коме рачунари откривају како да на основу великог броја података, које им дамо на располагање, генерализују некакве специфичности проблема које треба решити. Крајњи циљ јесте да се рачунарима омогући и поспеши процес аутоматског учења, али и доношење самосталних одлука без интервенције од стране човека. Генерализација, у овом смислу, јесте сама способност рачунара да даје тачне резултате над скупом података који није видео.

Подаци које шаљемо на улаз алгоритама машинског учења приликом обучавања још се називају и подаци за обучавање (енг. *Training data*). Они представљају основу за стицање искуства, тј. основу за генерализацију од стране алгоритама машинског учења.

Спектар примене ових алгоритама је веома широк и велика је могућност да их свакодневно користимо и долазимо у додицај са њима, а да тога нисмо свесни. Алгоритми машинског учења могу се наћи као саставни део виртуалних личних асистената, могу се пронаћи као део предикција које користе системи за навигацију, у системима видео надзора, друштвеним мрежама (људи које можда познајете, препознавање лица, ...), у филтрирању електронске поште, у онлајн корисничким подршкама, онлајн детекција превара, итд.

Како је спектар примене ових алгоритама широк, тако се типови алгоритама машинског учења разликују у свом приступу учењу, типу података које примају на улазу и које дају на излазу, па и у самом типу проблема које је потребно да реше.

У наредних неколико подналова дат је кратак опис неколико различитих приступа учења алгоритама машинског учења.

### **2.1.1 Надгледано учење**

Надгледано учење представља облик стицања искуства од стране алгорита машинског учења у коме се креира математички модел проблема на основу улазних података и жељених излазних података. Скуп података за обучавање садржи један или више улаза и жељени излаз, који се још назива и надгледајући сигнал.

Итеративном оптимизацијом функције циља долази се до функције којом алгоритам машинског учења може ефективно и поуздано да одреди излаз на основу улаза који до тада још није видео.

У типове надгледаног учења спадају: активно учење, класификација и регресија. Током активног учења алгоритам има способност да интерактивно тражи од корисника да означи податке како би се знао жељени излаз. Класификациони алгоритми се користе када излази имају ограничен број излазних вредност, док се алгоритми регресије користе када излази могу да узму било коју нумеричку вредност из неког интервала.

Овакав приступ учењу захтева велику количину података за обучавање, што је често временски скупо. [5]

### 2.1.2 Ненадгледано учење

Ненадгледано учење се базира на учењу уз помоћ скупа за обучавање који се састоји само од улаза, без назначених очекиваних излазних вредности. Овакви алгоритми уочавају заједничке структуре у подацима или груписање разних тачака и реагују на основу присуства или одсуства таквих заједничких особина. Мана му је што не може тачно да групише непознате податке. [5]

Главна примена овакве врсте учења јесте у процени густине у статистици, као што је налажење функције вероватноће густоће.

### 2.1.3 Полунадгледано учење

Полунадгледано учење јесте комбинација надгледаног учења и ненадгледаног учења. Комбинацијом ова два имамо податке за обучавање који садрже мањи број означених података, односно податке које садрже очекиване излазне вредности и податке који садрже само улазне вредности без података о излазним вредностима.

Овај метод учења предложен је од стране истраживачке заједнице као одговор на недостатке сваког од појединачних типова учења. [5]

## 2.2 Дубоко учење

Дубоко учење произлази из шире фамилије метода машинског учења у којима се, као носиоци процеса машинског учења, примењују вештачке неуронске мреже. Вештачке неуронске мреже инспирисане су природним биолошким процесима преноса и обраде информација у биолошким неуронским мрежама унутар животињског мозга. Те неуронске мреже састављене су од више слојева вештачких неурона, који су на одговарајући начин повезани. Сваки слој састоји се од различитог броја вештачких неурона од којих сваки врши некакву линеарну или нелинеарну математичку манипулацију сигнала који долази до њега. Због такве слојевите структуре вештачких неуронских мрежа долази придев „дубоко“.

Као и код машинског учења, крајњи циљ дубоког учења јесте да се омогући и поспешити рачунарима да, на основу улазних података, генерализују специфичности некаквог проблема. Уз помоћ генерализације тих специфичности циљ је да самостално и исправно донесу одлуку о решењу проблема који није виђен током њиховог обучавања, а без интервенције од стране човека. Такође и неки приступи таквом учењу су им заједнички: надгледано учење, ненадгледано учење и полунадгледано учење. Код



дубоког учења постоји и учење условљавањем (енг. *Deep Reinforcement Learning*), као још један од приступа обучавања вештачких неуронских мрежа. [6]

Као главна разлика између традиционалног машинског учења и дубоког учења наводи се начин на који се извлаче специфичности везане уз некакав проблем. [7] Оно што је специфично за дубоко учење јесте да процес учења пролази кроз слојеве вештачких неуронских мрежа, где сваки дубљи слој учи комплексније специфичности од претходног. На пример, у пољу рачунарске визије, унутар конволутивних неуронских мрежа, први слој може да врши извлачење, односно детекцију ивица, а последњи слој да детектује да ли је на слици слон или миш.

Оно што дубоко учење чини занимљивим јесте да може бити примењено у скоро сваком домену. Специфичности некаквог проблема не морају бити унапред пројектоване, већ се оне аутоматски уче. Исти приступ учењу може да се примени на различите проблеме, што може бити од велике користи код проблема који немају довољно расположивих података за учење.

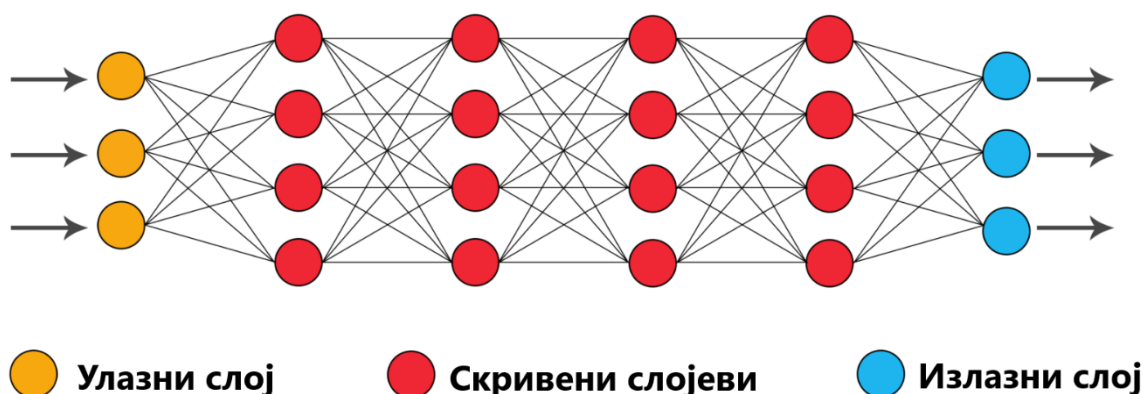
### 2.2.1 Вештачке неуронске мреже

Структура дубоких неуронских мрежа је таква да сигнал пролази кроз више слојева на свом путу од улаза до излаза из мреже. Сигнал се између слојева креће преко веза које повезују неуроне различитих слојева, а још се називају и синапсе. Сваки неурон једног слоја може бити повезан са сваким неуроном следећег слоја и тада такве слојеве називамо потпуно повезаним слојевима (енг. *Fully connected layers*). Вештачка неуронска мрежа сматра се „дубоком“ ако постоје бар 3 слоја кроз која сигнал пролази, укључујући и улазни и излазни слој. Оне могу имати велики број слојева, нпр. *ResNet* са 1202 слоја. [8]

Првобитна идеја јесте да вештачке неуронске мреже проблеме решавају на сличан начин као што их решава људски мозак, међутим касније је дошло до фокусирања на специфичне менталне способности, па је тако развијен приступ који пропушта сигнал кроз неуронску мрежу уназад како би се она наштимала за потребе улазних информација.

Вештачке неуронске мреже пројектоване су да препознавају заједничке особине сигнала из стварнога света из перспективе машине. Заједничке особине које оне препознавају су нумеричког типа и сви подаци из стварнога света, које желимо да пропустимо кроз вештачку неуронску мрежу, морају да буду представљени нумерички.

Често, да би се избегле неправилности у раду вештачке неуронске мреже, нумерички подаци представљени су у нормализованом опсегу вредности  $[0, 1]$  или  $[-1, 1]$ .



Слика 2.1 Структура дубоке неуронске мреже [9]

Сваки слој унутар мреже обучава се на посебним специфичностима сигнала у односу на излаз претходног нивоа. Што се улази дубље у мрежу, то су неурони способнији да препознају комплексније специфичности сигнала јер гомилају и рекомбинују специфичности претходног слоја.

Ако се ради о проблему класификације, проласком улазног сигнала кроз мрежу, на излазу се појављује вероватноћа некаквог излаза. На пример, ако пропустимо слику на којој се налази пас кроз, за то, обучену неуронску мрежу, она на излазу може дати вредност која представља вероватноћу да је на слици одређена пасмина.

Ако се ради о проблему регресије, проласком улазног сигнала кроз мрежу, на излазу се појављује нумеричка вредност која одговара нечему што би могло идуће да се деси, појави,... На пример, ако на улаз вештачке неуронске мреже доведемо информације о стамбеном простору, као што су: површина стамбеног простора, површина подрума, постоји ли двориште, клима, итд.; она на излазу може да предвиди његову цену.

Пресек области вештачке интелигенције, машинског учења, вештачких неуронских мрежа и дубоког учења дат је на слици 2.2.



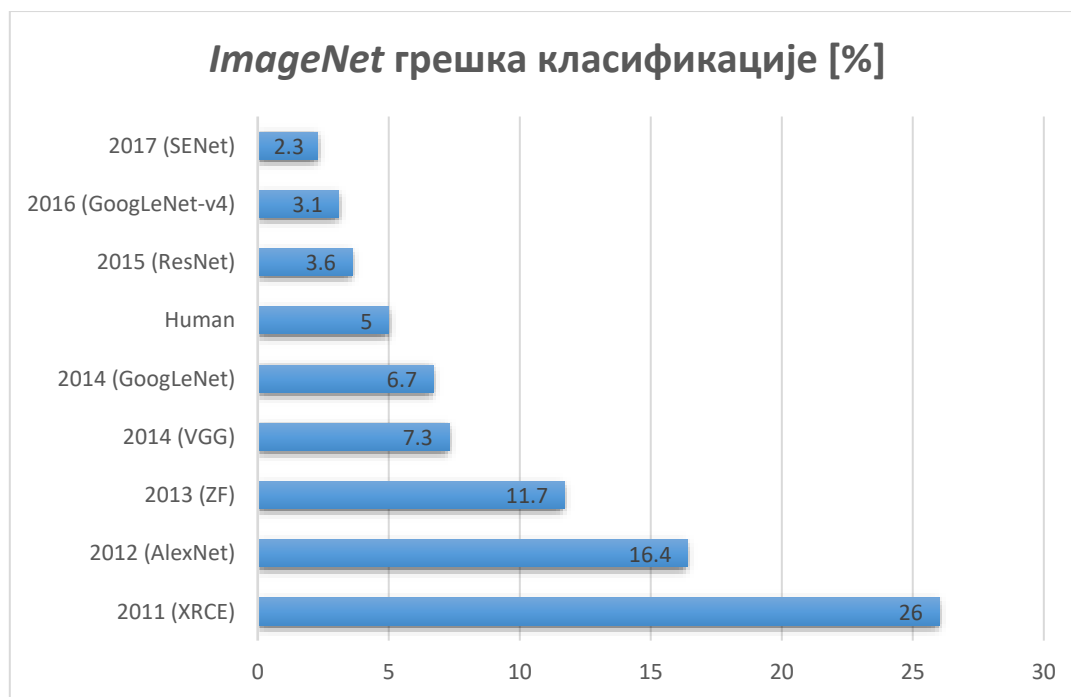
Слика 2.2 Структура поља вештачке интелигенције према [7], [10]

Архитектуре вештачких неуронских мрежа долазе у разним облицима, због различитих потреба и врста проблема. Нама најзначајнија архитектура, која се користи приликом израде овог рада јесте архитектура конволутивних неуронских мрежа. Кратак теоријски преглед описан је у секцији 2.3

### 2.2.2 Примене дубоког учења

У наредном делу дат је кратак преглед неких примена дубоког учења, од којих класификација фотографија и аутоматско препознавање говора имају значајан ниво перформансе. [11], [12]

Класификација фотографија на бази дубоког учења последњих је година доведена до завидног нивоа чак и за човека и човекове способности. Последњи резултати такмичења које евалуира алгоритме за детекцију и класификацију улазних фотографија (енг. *Large Scale Visual Recognition Challenge (ILSVRC)*) из 2017. године показују да је грешка класификације објеката алгоритма дубоког учења 2,3%, док је грешка код класификације објеката на фотографијама код човека 5%. [11]



Графикон 2.1 Грешка класификације на основу *ImageNet* скупа података према [11]

Дубоко учење нашло је своју примену и у препознавању говора и обично се врши на ТИМИТ акустично-континуалном говорном скупу података (енг. *TIMIT Acoustic-Phonetic Continuous Speech Corpus*), а грешка је одређена ПЕР постотком (енг. *Phone error rate*). Једно од бољих решења над ТИМИТ скупом података предложено је унутар рада [12], са ПЕР грешком од 16.9%.

Дубоке неуронске мреже су се показале као добар алат за идентификацију стилског периода дате фотографије, као и екстракцију датог стила и његову примена над насумичном фотографијом. Такође, у могућности су да генеришу занимљиве фотографије на основу насумичних визуалних улазних параметара.



Слика 2.3 *deepart.io* у комбинацији са Ван Гоговом Звезданом ноћи

Постоји и *AtomNet* систем дубоког учења за структурно пројектовање лекова. Кориштен је за предвиђање кандидата биомолекула за болести као што је Ебола и Мултипла склероза. [13]

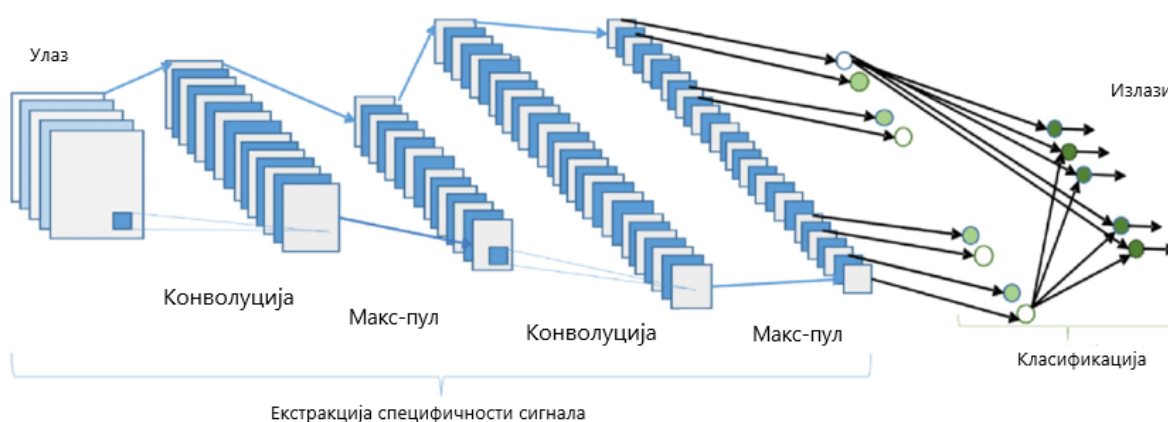
## 2.3 Конволутивне неуронске мреже

Конволутивне неуронске мреже представљају архитектуру вештачких неуронских мрежа које се најчешће примењују у пољу рачунарске визије. Једна од њихових битнијих карактеристика јесте да су транслационо инваријантне, односно имају способност да детектују исти објекат на различитим фотографијама без обзира на његов положај унутар фотографије. Оне су, такође, врста алгорита дубоког учења који је способан да на улазној слици одреди важност њених различитих делова и има могућност да диференцира те делове. Да би се вештачка неуронска мрежа сматрала конволутивном мрежом, потребно је да бар један скривени слој имплементира математичку операцију конволуције.

Инспирација за конволутивне неуронске мреже долази из природних биолошких процеса, па тако образац повезаности вештачких неурона у њима веома сличи обрасцу повезаности неурона животињског визуалног кортекса. Индивидуални неурони

одговарају на стимуланс само унутар ограничене регије визуалног поља, рецептивног поља (енг. *Receptive Field*). Рецептивна поља различитих неурона се деломично поклапају и тако покривају читаво визуално поље.

На слици 2.5 може се видети уобичајена архитектура конволутивне неуронске мреже. Два главна дела од којих се састоји свака конволутивна неуронска мрежа су: екстракција специфичности и класификација. Ако погледамо дубље унутар дела за екстракцију специфичности можемо приметити да конволутивна неуронска мрежа имплементира два типа слојева: конволутивни (енг. *Convolution*) и макс-пул (енг. *Max-pooling*) сажимајући слој. Сви ови слојеви, који се налазе између улаза и излаза неуронске мреже су скривени слојеви. Као активацијска функција, најчешће се користи РЕЛУ (енг. *RELU, Rectified linear unit*) слој који је попраћен додатним сажимајућим слојевима, потпуно повезаним слојевима и слојевима за класификацију. Активацијски слој, односно активацијска функција омогућава неуронској мрежи да разлучи које информације су од значаја, док сви ирелевантни подаци остају неискориштени. Детаљнији описи појединачних слојева и њихових функционалности могу се наћи у наредним поглављима.

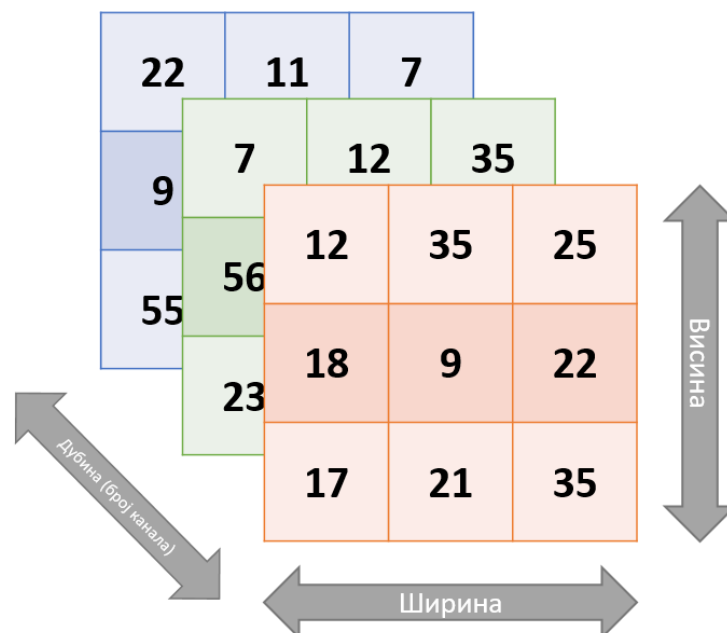


Слика 2.4 Пример архитектуре конволутивне неуронске мреже [14]

### 2.3.1 Улаз у мрежу

Ако се ради о слици РГБ (енг. *Red, Green, Blue* - *RGB*) формата улаз у конволутивну неуронску мрежу јесте слика димензија ( $W \times H \times C$ ), а која је представљена матрицом истих димензија. Где су:

- $W$  – висина
- $H$  – ширина
- $C$  – број канала (за РГБ формат – 3)

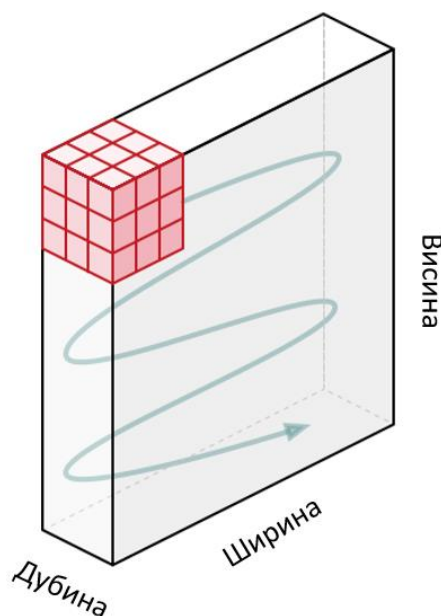


Слика 2.5 Преглед РГБ канала

Задатак конволутивне неуронске мреже јесте да слике сведе у форму која је једноставнија за обраду, али без губитака специфичности слике које су неопходне за давање добре предикције на излазу.

Формати простора боја, који се могу наћи на улазу у мрежу, су: црно-бела фотографија (енг. *Grayscale*), РГБ (енг. *Red, Green, Blue* - *RGB*), ХСВ (енг. *Hue, Saturation, Value* - *HSV*), итд.

### 2.3.2 Конволутивни слој



Слика 2.6 Приказ операције конволуције [15]

Са слике 2.7 се може видети операција конволуције која се врши над улазном сликом и кернелом (на слици означено црвеном бојом) чији су елементи параметри које је потребно да мрежа научи током процеса њеног обучавања. Кернел се по слици, у зависности од корака (енг. *Stride*), помера хоризонтално док не дође до руба, након чега се са истим кораком помера вертикално, ка доле, и враћа се на крајњу леву позицију. Поступак се понавља све док се не дође до краја матрице којом је слика представљена. Приликом сваког померања ради се матрични производ између кернела и дела матрице који је обухваћен тренутном позицијом кернела. У случају да се ради о сигналу као што је РГБ сигнал, са више канала, тада и кернел има исту дубину као и такав сигнал. Матрични производ се тада ради између сваког једнаког индекса дубине и врши се сумирање резултата са елементом помераја (енг. *bias*) како би добили једнодимензионални канал конволвираних специфичности сигнала.

Циљ операције конволуције јесте да из улазног сигнала генерише специфичности вишег нивоа. Конволутивни слојеви који се налазе ниже у мрежи (ближе улазу него излазу) детектују специфичности сигнала нижег нивоа, као што су: рубови, боје, итд. Како се крећемо дубље у мрежу (ка излазу), ти конволутивни нивои су обучени



да генералишу све комплексније специфичности улазног сигнала, али и разумевање комплетне слике.

Процес конволуције може се описати следећом формулом [14]:

$$x_j^i = f\left(\sum_{i \in M_j} x_i^{i-1} * k_{ij}^l + b_j^l\right) \quad (2.1)$$

где је  $x_j^i$  излаз из тренутног слоја,  $x_i^{i-1}$  излаз из претходног слоја,  $k_{ij}^l$  јесте кернел за тренутни слој,  $b_j^l$  представља елементе помераја тренутног слоја,  $M_j$  представља селекцију улазног сигнала. На крају излазни сигнал пролази кроз линеарну или нелинеарну активацијску функцију (сигмоид, хиперболични тангенс, софтмакс, РЕЛУ или функције идентитета).

За активацијску функцију најчешће се бира РЕЛУ, јер у поређењу са осталима позитивно утиче на перформансе мреже. Дефинише се формулом:

$$f(x) = \max(0, x) \quad (2.2)$$

а сврха јој је да у неуронску мрежу унесе нелинеарности.

Постоје две врсте конволуције:

- Истог проширења (енг. *Same Padding*) – ако сигнал слике величине  $5 \times 5 \times 1$  проширимо, односно додамо проширење, тако да проширени сигнал има димензије  $6 \times 6 \times 1$  и применимо кернел димензија  $3 \times 3 \times 1$ , добићемо излазни сигнал величине  $5 \times 5 \times 1$
- Тачног проширења (енг. *Valid Padding*) – ако на сигнал слике величине  $5 \times 5 \times 1$  применимо кернел димензија  $3 \times 3 \times 1$ , добићемо излазни сигнал димензија истих као што су димензије кернела,  $3 \times 3 \times 1$

Излазна димензија  $O$  из овог слоја може се израчунати као:

$$O = \frac{W - K + 2P}{S} + 1 \quad (2.3)$$

где је  $O$  излазна ширина или висина.  $W$  представља улазну ширину/висину,  $K$  величину филтера,  $P$  проширење (енг. *Padding*), а  $S$  корак (енг. *Stride*). Дубина излазног сигнала зависи од броја кориштених кернела.

Број параметара  $p$ , односно елемената кернела, које неуронска мрежа треба да изучи, може се за сваки слој израчунати на основу:

$$p = (N * M * L + 1) * k \quad (2.4)$$

где  $N$  представља ширину, а  $M$  висину кернела,  $L$  дубину улазног сигнала, а  $k$  дубину кернела. Такође, за сваки индекс дубине кернела додаје се по један елемент помераја који је потребно изучити.

### 2.3.3 Сажимајући слој

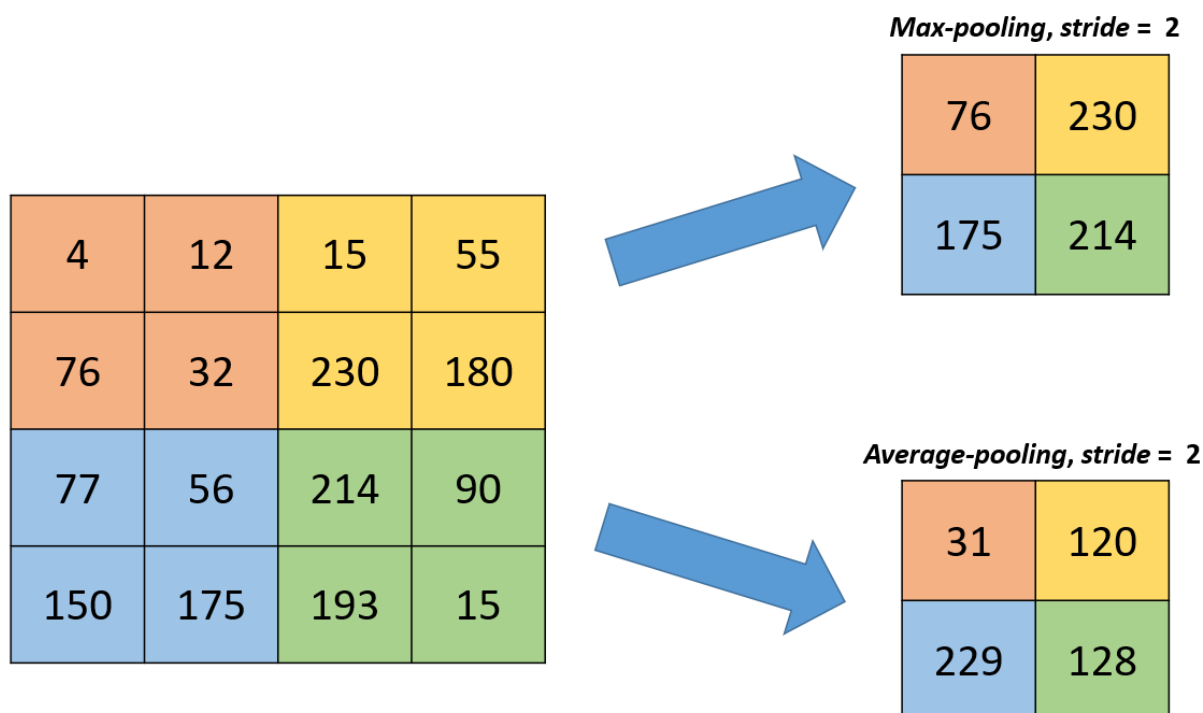
Сажимајући (енг. *Pooling*) слој има могућност смањивања просторне димензије сигнала у циљу смањења снаге потребне за израчунавање и обраду података. Такође је корисно за извлачење доминантних специфичности сигнала. Ако су димензије улазног сигнала  $(W \times H \times C)$ , димензије  $W$  (ширина) и  $H$  (висина) се мењају док димензија  $C$  (број канала или дубина) остаје непромењена. На пример, ако је кернел сажимајућег слоја димензија  $2 \times 2$ , ширина и висина улазног сигнала ће се смањити за пола.

Обично се користе један од следећа два приступа као сажимајући слој:

- *Max-pooling* – за излазну вредност узима се највећа вредности унутар дела улазног сигнала величине  $N \times N$ , где  $N$  представља димензионалност кернела сажимајућег слоја
- *Average pooling* – сумира се део величине  $N \times N$  улазног сигнала и проналази се средња вредност, која се узима за излазну вредност, где  $N$  представља димензионалност кернела сажимајућег слоја

Постоје и алтернативне методе сажимајућих слојева које су предложене, а неке од њих су: фракционални макс-пул (енг. *Fractional max-pool*), конволутивни пул (енг. *Sub-sampling with convolution*). [14]

Конволутивни слој и сажимајући слој заједно чине  $k$ -ти слој конволутивне неуронске мреже. Жртвовањем снаге која је потребна за извршавање рачунских операција, могуће је додавање већег броја оваквих слојева у конволутивну неуронску мрежу, како би она, у зависности од комплексности специфичности улазних слика, могла да генерализује све ситније детаље.



Слика 2.7 Пример *Max-pool* и *Average-pool* метода са кораком = 2

Димензије излазног сигнала након примене овог слоја, могу се израчунати по формули (2.3), при чему је корак  $S$  (енг. *stride*) једнак димензији  $K$  сажимајућег слоја. Пошто сажимајући слој узима улазни сигнал и „скупља“ га у сигнал мањих димензија, број параметара који је потребно да се изучи је једнак нули.

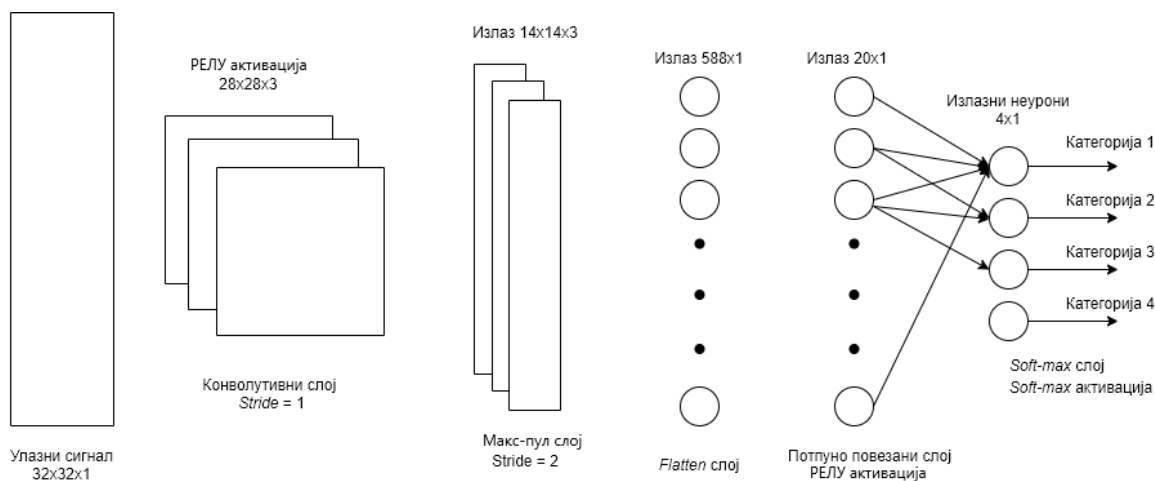
### 2.3.4 Потпуно повезани слој

Начин на који се могу научити специфичности сигнала које пролазе кроз конволутивне нивое унутар мреже јесте додавањем потпуно повезаног слоја. Излазни сигнал из конволутивних слојева, који је представљен матрицом, се исправља (енг. *Flatten*) у вектор колоне и такав пролази кроз потпуно повезани слој.

Улога потпуно повезаног слоја јесте да научи да интерпретира специфичности сигнала које долазе из конволутивних слојева и да на основу њих донесу закључак о проблему који се разматра.

Излази из потпуно повезаног слоја најчешће су вероватноће које одговарају некаквом проблему класификације. На пример, излази из овог слоја могу да дају информацију о томе да ли се на слици налази пас или мачка, да ли на слици постоји некакав објекат, итд. Исто тако, као излаз могу се појавити информације о позицији

објеката на слици. На пример, на излазу се може појавити вредност која даје информацију где се на слици налази центар лица или ока.



Слика 2.8 Потпуно повезани слој вештачких неуронских мрежа

Број тежинских коефицијената  $w$  овог слоја, које је потребно изучити добија се као:

$$w = (n + 1) * m \quad (2.5)$$

где  $n$  представља број улаза, а  $m$  број излаза. Такође сваком улазном тежинском коефицијенту придружује се по један елемент помераја.

### 2.3.5 Обучавање, параметри модела и хипер-параметри

Неуронске мреже одликују параметри и хипер-параметри модела. Обе врсте параметара су битне јер озбиљно утичу на њихове перформансе, али и на брзину обучавања. Параметри модела су они параметри које неуронска мрежа треба да научи приликом обучавања. Када причамо о конволутивним неуронским мрежама, у њих спадају параметри конволутивног слоја и тежински коефицијенти неурона у потпуно повезаним слојевима. Специфични су за модел мреже и њихове вредности се, у процесу обучавања, формирају на основу специфичности сигнала на улазу у мрежу. Хипер-параметре модела одабере руковалац неуронском мрежом и у њих спадају: функција оптимизације, активацијска функција неурона, број скривених слојева модела, итд. Хипер-параметри, као што је број скривених слојева модела, знатно утичу на рад неуронских мрежа и не постоји универзално правило по коме би се они одабрали, већ добар одабир зависи од искуства руковаоца мрежом, али и самог проблема који је потребно решити. Због тога, често се користе једна од три прихваћене методе од стране

заједнице људи који се баве обучавањем неуронских мрежа и њиховом применом у разним апликацијама.

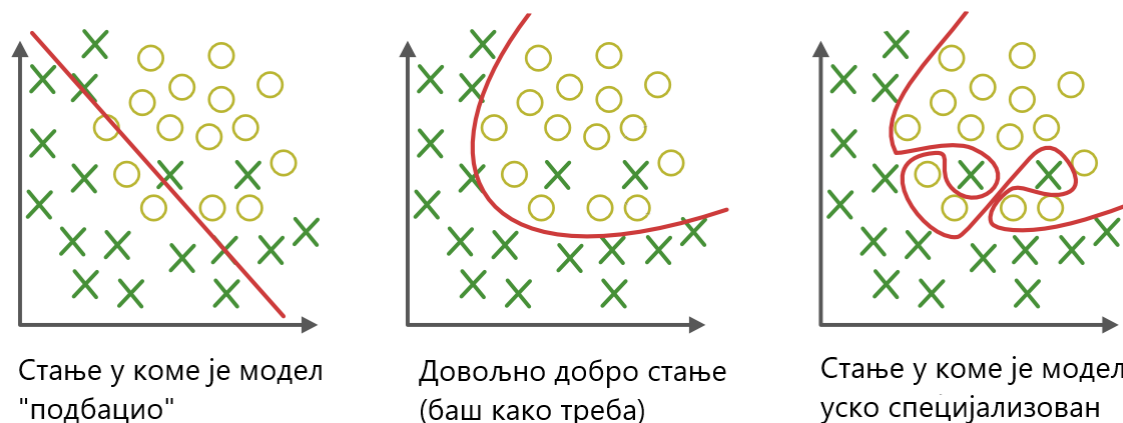
Прва метода одабира погодних хипер-параметара модела неуронских мрежа је итеративна метода, где се приликом сваке итерације обучавања врше погађања хипер-параметара модела, проверавају метрике тачности и разматрају се они хипер-параметри уз које је модел дао најбоље резултате. Ова метода назива се још и *Trial and error* метода. Друга метода одабира погодних хипер-параметара заснива се на системском тестирању више различитих вредности сваког хипер-параметра. Она се још назива и *Grid search* метода и може да буде веома спора за велики број хипер-параметара. Трећа метода је метода насумичног одабира. Она често даје резултате који нису интуитивни и често није јасно из ког разлога је одређени хипер-параметар погодно одабран. На основу [16] дошло се до закључка да је последња метода много ефикаснија од претходне две.

До закључка да ли су одабрани погодни хипер-параметри модела најчешће се долази уз помоћ метрике тачност валидације. Приликом анализирања метрике тачности валидације модел се може наћи у једном од три стања: стање у коме је модел „подбацио“, стање у коме је модел уско специјализован за скуп података за обучавање и довољно добро стање (баш како треба).

Уколико модел неуронске мреже „подбаци“, он неће давати жељене резултате над скупом података за обучавање, али ни над подацима које није видео приликом обучавања, односно неће имати никакву способност генерализације специфичности сигнала који је потребно обрадити. Ово стање се још назива *underfitting* и оно је нежељено стање модела неуронске мреже.

Уколико је модел неуронске мреже у стању у коме је уско специјализован за скуп података за обучавање, он ће давати изразито добре предикције над сетом података за обучавање, али ће имати јако лоше предикције над подацима које није видео приликом обучавања. Ово стање се још назива и *overfitting* и, као и претходно стање, оно је нежељено стање модела неуронске мреже.

Жељено стање у коме руковалац моделом неуронске мреже жели да се она нађе јесте довољно добро стање у коме модел може дати довољно добре резултате и над скупом података за обучавање и над подацима које није видео приликом обучавања. Да ли су резултати довољно добри зависи од проблема који се посматра и од спремности руковаоца мрежом да толерише одређени степен грешке.



Слика 2.9 Нежељена и жељена стања модела неуронских мрежа [17]

Хипер-параметри неуронских мрежа могу се поделити на оне који су везани за структуру саме мреже и на оне који утичу на ефикасност њеног обучавања.

У хипер-параметре који су везани за структуру неуронске мреже спадају: број скривених слојева, иницијализација вредности тежинских коефицијената, активацијска функција и фактор одбацивања неурона приликом обучавања. Одабир броја скривених слојева неуронских мрежа зависи од комплексности проблема који се посматра и додавање додатних скривених слојева, односно мењање броја скривених слојева, може поспешити тачност рада модела. Како би неурони имали способност уочавања различитих специфичности сигнала датог проблема и да би се избегла симетрија између тежинских коефицијената различитих неурона препоручљиво је да се њихове вредности иницијализују на случајан начин. Активацијска функција директно утиче на конвергенцију неуронске мреже и на брзину обучавања и овај хипер-параметар одређује која ће се активацијска функција користити за обраду улазног сигнала унутар сваког појединачног неурона. Фактор одбацивања неурона приликом обучавања утиче на то колики постотак неурона ће бити одбачен приликом сваке епохе обучавања како би се избегло стање у коме је модел неуронске мреже „подбацио“.

У хипер-параметре који утичу на ефикасност њеног обучавања спадају: брзина обучавања (енг. *Learning rate*), смањивање брзине обучавања (енг. *Learning rate decay*), алгоритми оптимизације, број епоха обучавања и величина групе података за обучавање (енг. *Batch size*). Избор превише мале вредности хипер-параметра брзине обучавања може довести до повећања брзине обучавања, али и машења минимума функције губитака (енг. *Loss function*) чији минимум се тражи како би се смањила грешка између

предикција мреже и очекиваних резултата скупа података за обучавање. Због тога, најпогодније је увести хипер-параметар смањења брзине обучавања који ће омогућити да се смањи време обучавања како се модел мреже приближава адекватном решењу. Како би се избегло поменуто заглављивање у локалном минимуму функције губитака уводи се хипер-параметар који дефинише који ће се алгоритам оптимизације користити приликом обучавања и његов одабир зависи од проблема који се проматра. Неки од њих су: *Stochastic Gradient Descent*, *Momentum*, *Adam* и *AdaDelta*, чији описи превазилазе оквири овог рада. Број епоха обучавања говори колико пута улазни подаци за обучавање пролазе кроз неуронску мрежу приликом обучавања и пожељно га је повећавати до броја када тачност валидације над скупом података за валидацију почне да опада, па чак и ако се деси да тачност валидације над сетом података за обучавање настави да расте. Ово се ради из разлога како би се избегло нежељено стање у коме је модел неуронске мреже „подбацио“. Величина групе података за обучавање одређује колики се број података из скупа података за обучавање паралелно доводи на улаз неуронске мреже у току једне епохе јер понекад довођење целог скупа података на улаз може бити немогуће или временски скупо.

Из свега приложеног у овом поглављу, можемо закључити да је процес обучавања вештачких неуронских мрежа комплексан и итеративан процес. На перформансе система који решава посматрани проблем утичу разни фактори, почев од оних комплекснијих, као што су структура неуронске мреже и одабир функције оптимизације, па све до оних једноставнијих као што је избор броја епоха обучавања.

### 3 Концепт решења

Оно што је нама људима интуитивно и што можемо са лакоћом и несвесно да обрадимо и процесуирамо, као што је разлучивање да ли се на слици налази лице, где су различити елементи тог лица и да ли је поглед усмерен право, у страну, или негде другде, па чак и да ли је тај неко са фотографије поспан, за рачунар представља озбиљан проблем. Да би одредили степен пажње човека (возача) потребно је посматрати његово лице и положаје лица, али исто тако посматрати и елементе на лицу, као што су очи, уста, уши, итд. На основу различитих научних радова [18], [19], највећу информацију о степену пажње човека, на његовом лицу носе очи. Према томе, процена нивоа пажње може ефикасно да се врши на основу праћења различитих стања и положаја очију кроз време. Савладавање ове препреке рачунар може постићи уз помоћ приступа са дубоким учењем и конволутивним неуронским мрежама. Да би омогућили боље и прецизније одређивање поменутих специфичности, у овом раду користе се три конволутивне неуронске мреже уз приступ надгледаног учења. Свака мрежа обавља део посла, респективно, проналазак лица на слици, одређивање елемената лица са слике лица и обраду елемената лица и давање информације о стању возача. Због тога је потребно генерисати и означити за сваку мрежу по један скуп података.

Решење проблема подељено је у две целине: генерисање и означавање скупова података за обучавање и структура система за праћење пажње возача.



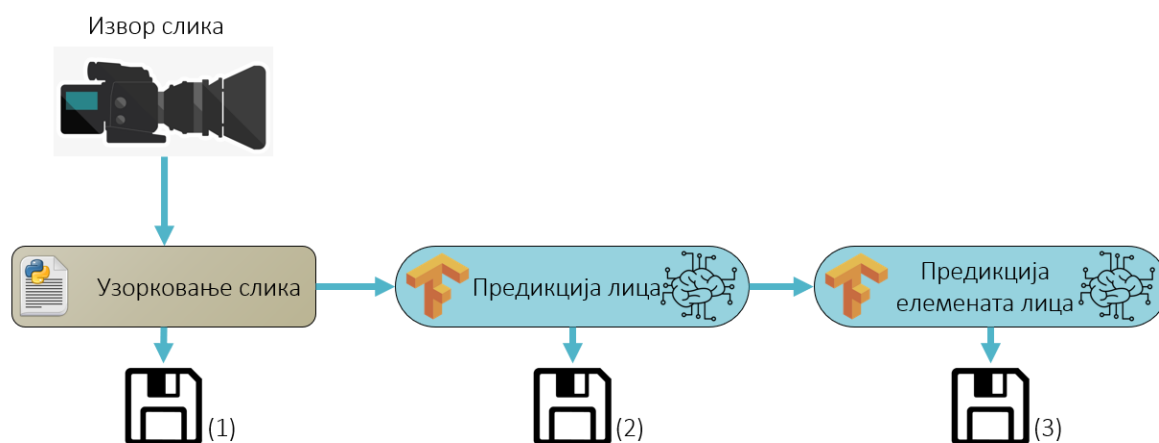
### 3.1 Генерисање и означавање скупова података

Да би неуронске мреже имале добру способност генерализације и уопште добру моћ предикције над подацима које још нису виделе, генерисање и означавање добрих скупова података је од великог значаја. Сами скупови података представљају слике које су, заједно са својом одговарајућом *.csv* (енг. *Comma-separated values*) датотеком, улази у одговарајућу неуронску мрежу у фази обучавања.

#### 3.1.1 Генерисање скупова података

Генерисање се може поделити у три нивоа:

- 1) Генерисање фотографија унутар возила – врши се из видео извора који може бити или веб камера или видео снимак уз помоћ *Python* скрипта и *OpenCV* библиотеке што омогућава узорковање и чување фотографија на рачунар.
- 2) Генерисање фотографија лица – врши се уз помоћ претходно обучене неуронске мреже која даје предикцију, да ли и где се лице на слици налази. На основу њених предикција са целе фотографије се исеца слика лица и похрањује на рачунар.
- 3) Генерисање фотографија елемената лица – врши се уз помоћ претходно обучене неуронске мреже која даје предикцију да ли и где се налазе очи лица на фотографији лица. На основу њених предикција са фотографије лица исецају се очи и слике се похрањују на рачунар.



Слика 3.1 Илустровани приказ тачке 1, 2 и 3, генерисање скупова података

Потребно је да фотографије унутар генерисаних скупова буду одређених димензија за сваки скуп, али и константних димензија унутар сваког скупа. Приказ димензија фотографија унутар сваког скупа дат је табеларно:

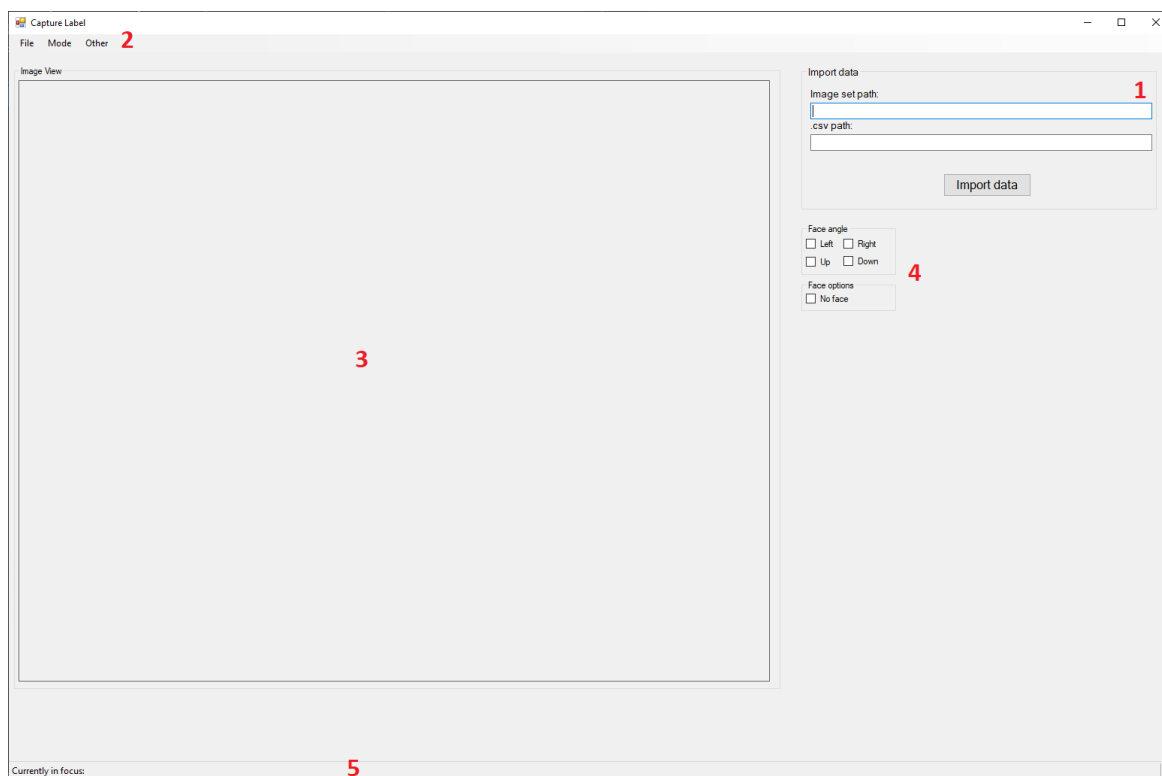
Скуп података	Резолуција [пиксел]
Унутар возила	640x480
Лица	300x200
Елементи лица	100x100

Табела 3.1 Резолуције генерисаних скупова података

### 3.1.2 Означавање скупова података

Као што је споменуто, решење проблема реализовано је уз помоћ скупа конволутивних неуронских мрежа и приступа надгледаног учења. Према томе, да би се обучиле неуронске мреже потребно је, уз сваки скуп података, имати и скуп очекиваних вредности на излазима мреже. Процес у коме се генеришу очекиване вредности и похрањују у .csv датотеке назива се означавање. Означавање се врши ручно, унутар сваког скупа података, за сваку слику појединачно.

Програмско решење означавања скупова података реализовано је као *Windows Forms Application (Microsoft .NET Framework 4.7.2)* уз помоћ C# програмског језика. Јединствено програмско решење омогућава нам означавање скупова података за сва три нивоа неуронских мрежа. Структура апликације може се видети на слици 3.2.



Слика 3.2 Структура апликације за означавање података

1. Увези податке (енг. *Import data*) група – представља улаз у апликацију и сачињена је од три елемента:
  - a. Путања до директоријума са фотографијама (енг. *Image set path*) – путања до директоријума у коме се налазе улазне фотографије за означавање
  - b. .csv путања (енг. *.csv path*) – путања до .csv датотеке (уколико постоји) асоциране са фотографијама које желимо да означимо. Поље је опционо.
  - c. Увези податке дугме (енг. *Import data*) – покреће процес увожења фотографија у апликацију за означавање на основу претходна два поља
2. Трака са падајућим изборницима – садржи неколико падајућих изборника:
  - a. Датотека (енг. *File*) – омогућава стандардне интеракције са апликацијом за означавање, као што су: ново, спреми, спреми као, извези нормализоване .csv датотеке и излаз из апликације
  - b. Режим рада (енг. *Mode*) – поседује информације о тренутном режиму рада апликације за означавање, али и омогућава избор режима рада
  - c. Друго (енг. *Other*) – омогућава увид у основне информације о апликацији за означавање
3. Простор за фотографије (енг. *Image view*) – простор за приказ тренутне фотографије која је у процесу означавања

4. Атрибути тренутне фотографије – атрибути који се асоцирају са тренутном фотографијом, а зависе од режима рада апликације за означавање
5. Статусна трака – даје информацију о индексу тренутне фотографије унутар директоријума са фотографијама (десно) и информацију о тренутно фокусираној тачки (лево)(енг. *Currently in focus*)

Да би се омогућило брже и ефикасније означавање фотографија, имплементирани су и практичне пречице на тастатури и мишу. Омогућено је листање слика унапред и уназад помоћу *Page Down* и *Page Up* тастера и уз помоћ тачкића миша. Имплементирана је могућност копирања координата правоугаоника претходне слике на текућу уз помоћ тастера *X*, али и фино кориговање позиције правоугаоника уз помоћ стрелица на тастатури за по један пиксел. Такође, омогућена је брза промена правоугаоника, у зависности од режима рада, уз помоћ тастатуре, тастерима: 1, 2, 3, *q*, *w*, *e*.

Потребно је напоменути да апликације за означавање не мења улазне резолуције фотографија иако се то на први поглед може учинити. Привидне промене резолуције су естетске и практичне природе, те није потребно водити рачуна о резолуцијама фотографија унутар апликације за означавање. Узимајући то у обзир апликација за означавање током свог рада не чува стварне вредности координата, већ релативне у односу на привидну промену резолуције. Тек када се од њега потражи да се означене вредности сачувају у излазну *.csv* датотеку оне се прерачунавају у стварне координате. Такође, излазне датотеке могу се користити и као улазне датотеке у њега самог уношењем путање у поље *.csv* путања (енг. *.csv path*) и апликација за означавање тада прерачунава стварне координате у релативне.

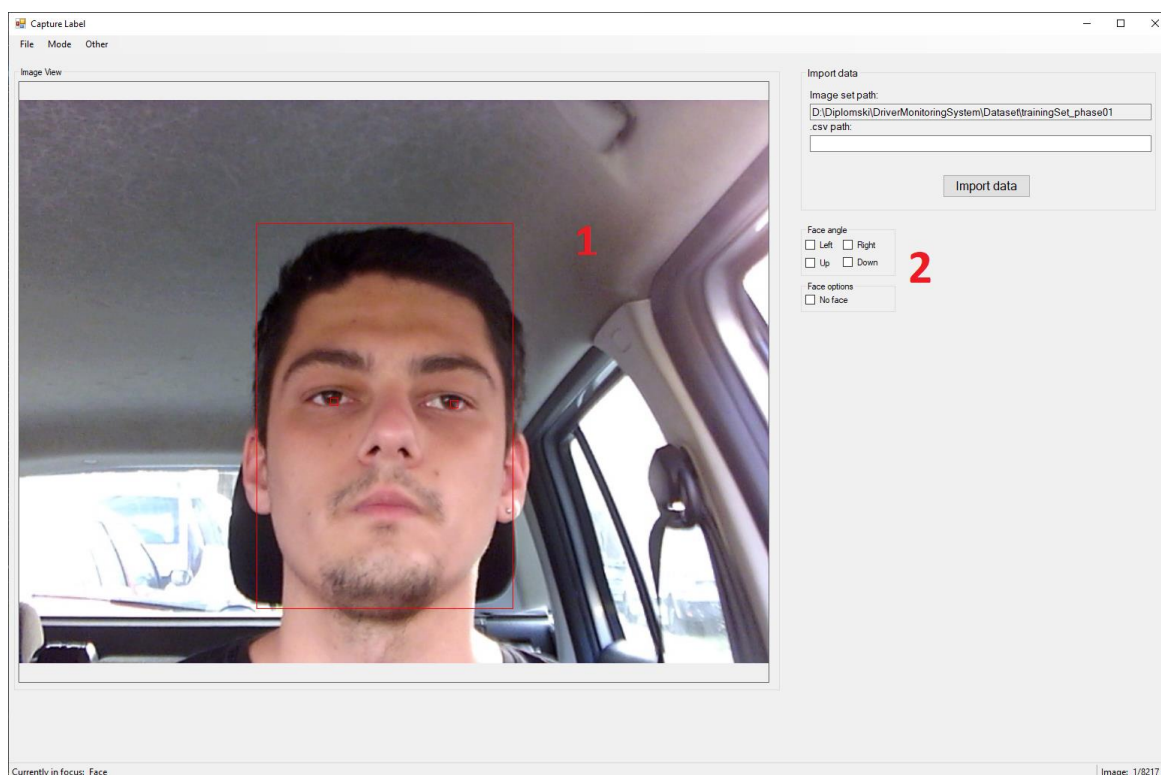
У наредних неколико потпоглавља описани су режими рада и излазне датотеке програмског решења за означавање скупова података.

### 3.1.2.1 Режим рада откривања лица

Режим рада откривања лица (енг. *Face detection*) карактеришу две ствари:

1. Простор за фотографије (енг. *Image view*) – простор за приказ тренутне фотографије која је у процесу означавања. Три правоугаоника служе за одређивање тачака од интереса. Помоћу највећег фиксира се централна тачка лица, остала два служе за фиксирање централних тачака очију. Величина највећег правоугаоника је промењива и константног омера ширине и висине 2:3

2. Атрибути тренутне фотографије – атрибути који се асоцирају са тренутном фотографијом су:
  - a. Страна у коју је лице усмерено (енг. *Face angle*) – очекиване вредности су: лево или десно, горе или доле. Ако нити једно поље није означено закључак је да је лице усмерено право
  - b. Опције лица (енг. *Face options*) – вредност поља „нема лица“ (енг. *No face*) носи информацију да ли на слици уопште постоји лице. Ако је означено лице на слици не постоји.



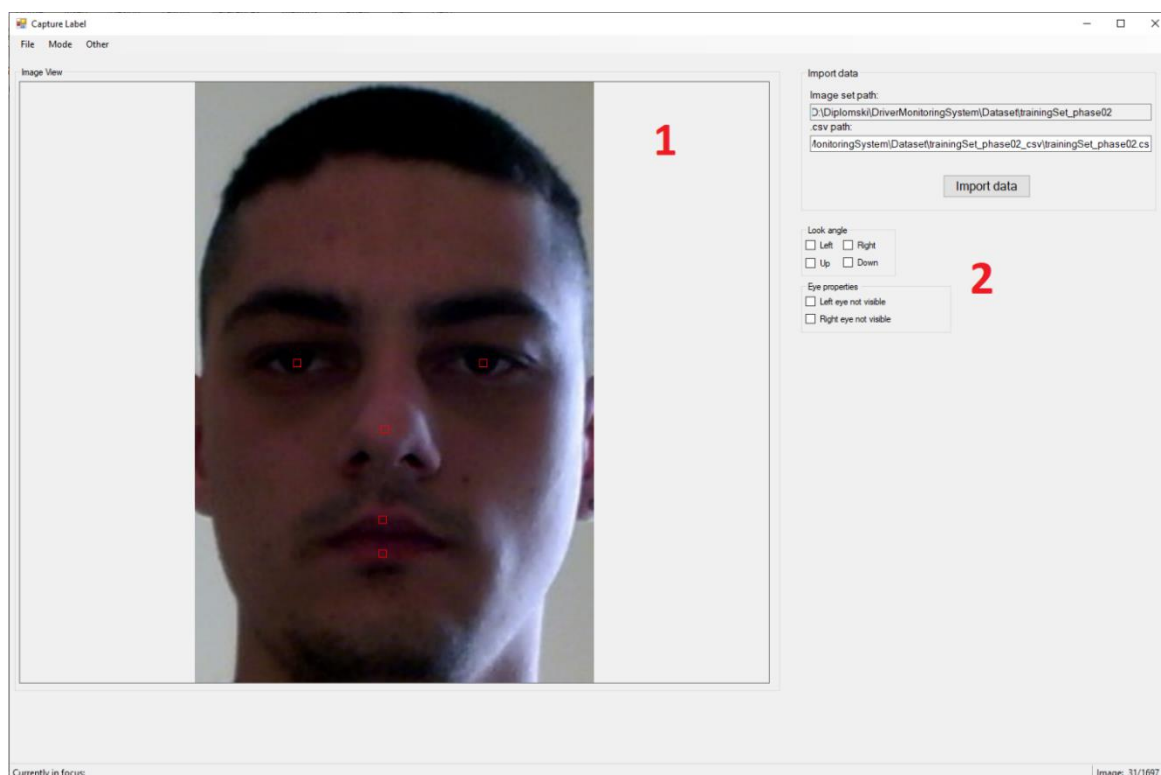
Слика 3.3 Режим рада откривања лица

### 3.1.2.2 Режим рада откривања елемената лица

Режим рада откривања елемената лица (енг. *Face elements detection*) карактеришу две ствари:

1. Простор за фотографије (енг. *Image view*) – простор за приказ тренутне фотографије која је у процесу означавања. Пет правоугаоника служи за фиксирање тачака од интереса: центар левог ока, центар десног ока, центар носа, горња усна, доња усна
2. Атрибути тренутне фотографије – атрибути који се асоцирају са тренутном фотографијом су:

- a. Правац у коме су очи усмерене (енг. *Look angle*) – очекиване вредности су: лево или десно, горе или доле. Ако нити једно поље није означено закључак је да су очи на лицу усмерене право.
- b. Специфичности очију (енг. *Eye properties*) – очекиване вредности су: лево око није видљиво (енг. *Left eye not visible*) или десно око није видљиво (енг. *Right eye not visible*) . Ако нити једно поље није означено, оба ока су видљива.



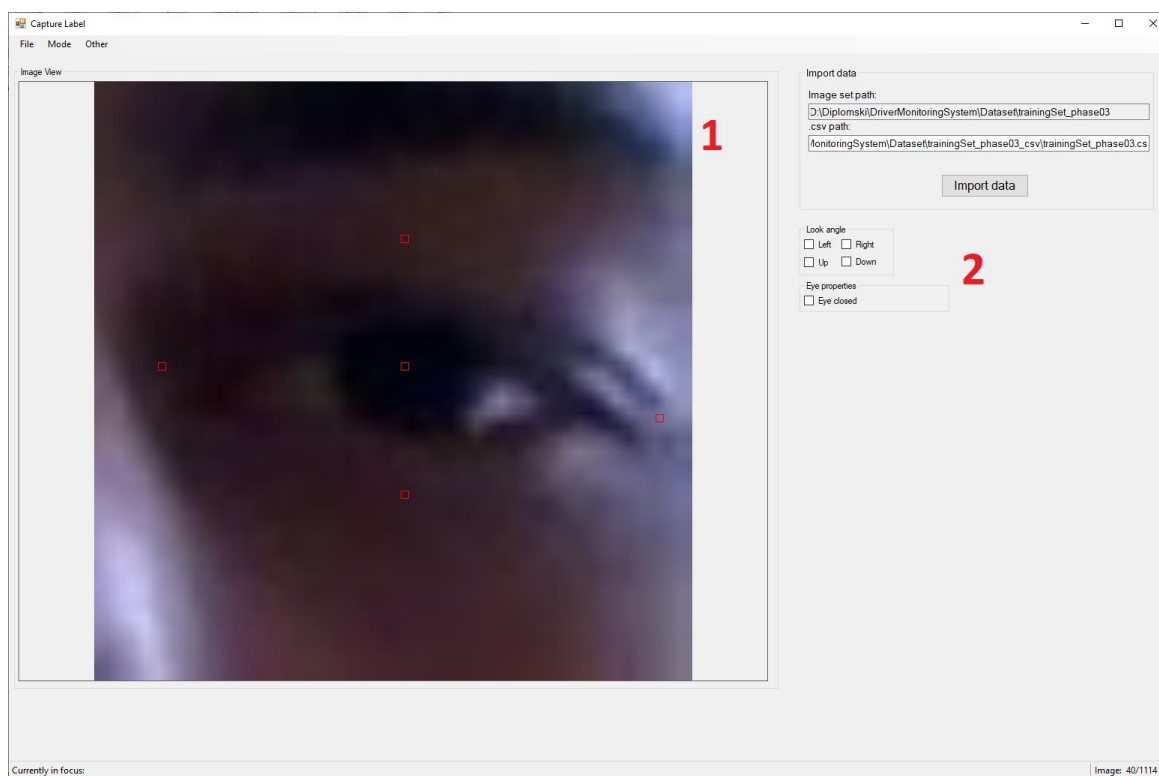
Слика 3.4 Режим рада откривања елемената лица

### 3.1.2.3 Режим рада откривања контура ока

Режим рада откривања контура ока (енг. *Eye contour detection*) карактеришу две ствари:

1. Простор за фотографије (енг. *Image view*) – простор за приказ тренутне фотографије која је у процесу означавања. Пет правоугаоника служи за фиксирање тачака од интереса: центар зенице, леви, десни, горњи и доњи руб ока.
2. Атрибути тренутне фотографије – атрибути који се асоцирају са тренутном фотографијом су:

- a. Правац у коме су очи усмерене (енг. *Look angle*) – очекиване вредности су: лево или десно, горе или доле. Ако нити једно поље није означено закључак је да су очи усмерене право.
- b. Специфичности очију (енг. *Eye properties*) – вредност поља „око је затворено“ (енг. *Eye closed*) носи информацију да ли је око отворено. Ако је ово поље означено, значи да је око затворено.



Слика 3.5 Режим рада откривања контура ока

#### 3.1.2.4 Излазне датотеке апликације за означавање

У зависности од изабраног режима рада програмског решења за означавање података, генеришу се по три излазне *.csv* датотеке за сваки режим рада. Ове датотеке садрже зарезом одвојене вредности. Оне су погодне и веома практичне за даље коришћење и припремање података за улаз неуронских мрежа. Свака врста у њима носи информације о једној слици унутар скупа података.

Прва *.csv* датотека чува информацију о координатама тачака од интереса појединачних слика унутар сваког скупа података, друга чува вредности координата у нормализованом опсегу  $[0, 1]$ , а трећа представља информације које су потребне за нормализацију и денормализацију података. Пошто се ради о информацијама на фотографијама унутар некаквог директоријума, прва и друга *.csv* датотека имају исти

број врста и константан број колона. Трећа .csv датотека се састоји од две врсте и онолико колона колико их одређује изабрани режим рада. Такође свакој врсти придружене су и целобројне представе атрибута те фотографије.

Пошто се, приликом обучавања, улази у неуронску мрежу представљају подацима у нормализованом опсегу вредности,  $[-1, 1]$  или  $[0, 1]$  апликација за означавање има улогу да нормализује претходно означене податке. У нашем случају улази су у опсегу вредности  $[0, 1]$  и нормализују се по формули:

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}}, \quad (3.1)$$

где  $X_n$  представља нормализовану вредност,  $X$  вредност тачке у пикселима, а  $X_{min}$  и  $X_{max}$  минималну и максималну вредност једне колоне унутар прве излазне датотеке.

У наредном делу дат је преглед поља свих .csv датотека

Преглед поља прве и друге .csv датотеке за режим рада откривања лица:

1. *Picture* – назив фотографије унутар директоријума
2. *noFace* – информација да ли се на слици налази лице (0 – налази, 1 – не налази)
3. *Face (x,y)* – позиција центра лица
4. *Left eye (x,y)* – позиција центра левог ока
5. *Right eye (x,y)* – позиција центра десног ока
6. *Left* - лице усмерено ка лево (0 – није, 1 – јесте)
7. *Right* – лице усмерено ка десно (0 – није, 1 – јесте)
8. *Up* - лице усмерено ка горе (0 – није, 1 – јесте)
9. *Down* – лице усмерено ка доле (0 – није, 1 – јесте)
10. *Face width* – ширина лица у пикселима

Преглед поља прве и друге .csv датотеке за режим рада откривања елемената лица:

1. *Picture* – назив фотографије унутар директоријума
2. *noLeftEye* – информација да ли се на слици налази лево око (0 – налази, 1 – не налази)
3. *noRightEye* – информација да ли се на слици налази десно око (0 – налази, 1 – не налази)
4. *Left eye (x,y)* – позиција центра левог ока на слици
5. *Right eye (x,y)* – позиција центра десног ока на слици



6. *Nose* ( $x, y$ ) – позиција центра носа на слици
7. *Mouth Up* ( $x, y$ ) – позиција горње усне
8. *Mouth Down* ( $x, y$ ) – позиција доње усне
9. *Left* - зеница ока усмерено ка лево (0 – није, 1 – јесте)
10. *Right* – зеница ока усмерено ка десно (0 – није, 1 – јесте)
11. *Up* - зеница ока усмерено ка горе (0 – није, 1 – јесте)
12. *Down* – зеница ока усмерено ка доле (0 – није, 1 – јесте)
13. *Face width* – ширина лица у пикселима

Преглед поља прве и друге .csv датотеке за режим рада откривања контура ока:

1. *Picture* – назив фотографије унутар директоријума
2. *eyeClosed* – информација да ли је око затворено (0 – није, 1 – јесте)
3. *Center Up* ( $x, y$ ) – позиција централне горње тачке ока
4. *Center* ( $x, y$ ) – позиција центра зенице
5. *Center Down* ( $x, y$ ) – позиција централне доње тачке ока
6. *Left Point* ( $x, y$ ) – позиција леве тачке ока
7. *Right Point* ( $x, y$ ) – позиција десне тачке
8. *Left* - зеница ока усмерено ка лево (0 – није, 1 – јесте)
9. *Right* – зеница ока усмерено ка десно (0 – није, 1 – јесте)
10. *Up* - зеница ока усмерено ка горе (0 – није, 1 – јесте)
11. *Down* – зеница ока усмерено ка доле (0 – није, 1 – јесте)
12. *Eye width* – ширина ока у пикселима

Преглед поља треће .csv датотеке за режим рада откривања лица:

1. *Min* – Врста која представља минималне вредност одређене колоне
2. *Max* – Врста која представља максималне вредност одређене колоне
3. *Face* ( $x, y$ ) – Минималне и максималне вредности координата центра лица свих фотографија
4. *Left eye*( $x, y$ ) - Минималне и максималне вредности координата левог ока свих фотографија
5. *Right Eye* ( $x, y$ ) - Минималне и максималне вредности координата десног ока свих фотографија
6. *Face width* - Минималне и максималне вредности ширине лица свих фотографија

Преглед поља треће .csv датотеке за режим рада откривања елемената лица:

1. *Min* – Врста која представља минималне вредност одређене колоне
2. *Max* – Врста која представља максималне вредност одређене колоне
3. *Left eye(x,y)* - Минималне и максималне вредности координата левог ока свих фотографија
4. *Right Eye (x,y)* - Минималне и максималне вредности координата десног ока свих фотографија
5. *Nose (x,y)* - Минималне и максималне вредности координата носа свих фотографија
6. *Mouth Up (x,y)* - Минималне и максималне вредности координата горње усне свих фотографија
7. *Mouth Down (x,y)* – Минималне и максималне вредности координата доње усне свих фотографија

Преглед поља треће .csv датотеке за режим рада откривања контура ока:

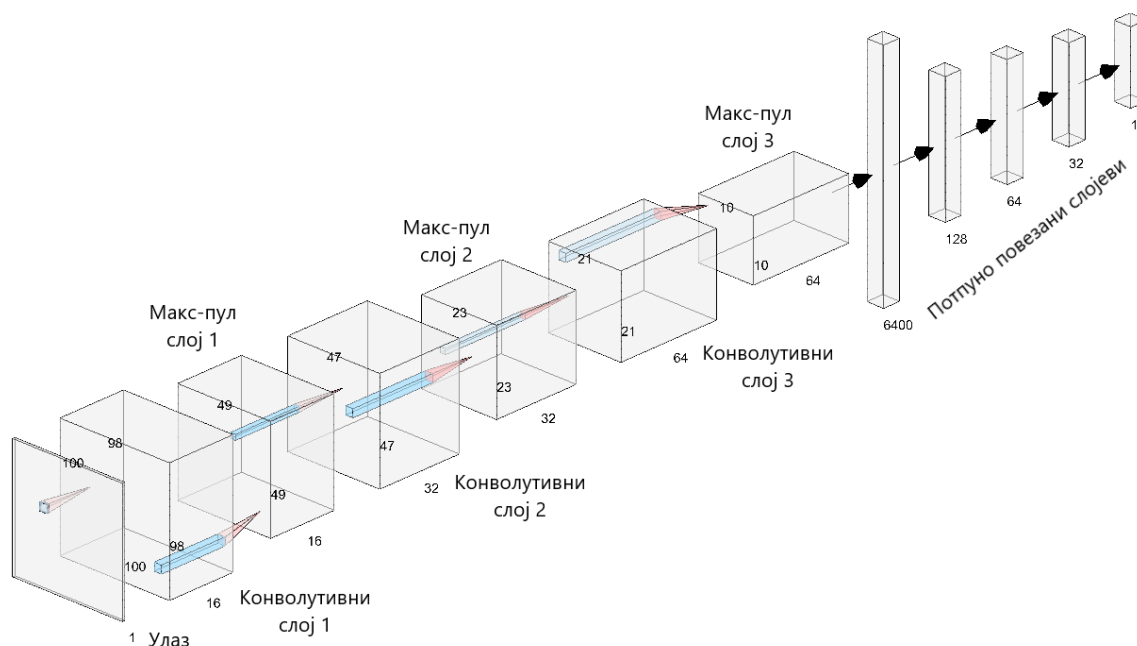
1. *Min* – Врста која представља минималне вредност одређене колоне
2. *Max* – Врста која представља максималне вредност одређене колоне
3. *Center Up (x,y)* – Минималне и максималне вредности координата централне горње тачке ока свих фотографија
4. *Center (x,y)* - Минималне и максималне вредности координата центра зенице ока свих фотографија
5. *Center Down (x,y)* - Минималне и максималне вредности координата централне доње тачке ока свих фотографија
6. *Left Point (x,y)* - Минималне и максималне вредности леве тачке ока свих фотографија
7. *Right Point(x,y)* – Минималне и максималне вредности десне тачке ока свих фотографија

## 3.2 Апликација за праћење пажње возача

У овом поглављу биће описани модели конволутивних неуронских мрежа кориштених за формирање апликације за праћење пажње возача, процес обучавања мрежа и структура финалне апликације. Апликација се састоји од три неуронске мреже од којих свака има посебан задатак и улогу у одређивању степена пажње присутне или одсутне код возача.

### 3.2.1 Модели конволутивних неуронских мрежа

Као што је споменуто, у имплементацији решења за различите ствари користе се три конволутивне неуронске мреже. Прва мрежа на улазној фотографији тражи лице и централне тачке очију, друга са фотографије лица тражи централне тачке очију, носа и горње и доње усне. Трећа мрежа потражује леву, десну, горњу и доњу рубну тачку ока, али и центар зенице. За све три мреже кориштена је слична архитектура, где разлику чини број излаза из сваке. Детаљна архитектура приказана је на слици 3.6.



Слика 3.6 Архитектура конволутивне неуронске мреже [20]

Улаз у мрежу чини црно-бела фотографија димензија 100x100 пиксела, која пролази кроз први конволутивни слој чији је кернел димензија (16, 3, 3), након тога користи се макс-пул сажимајући слој димензија (2,2). Такав сигнал преузима други конволутивни слој са кернелом димензија (32, 3, 3) и поновно се користи макс-пул

сажимајући слој димензија (2,2). Након тога, сигнал пролази кроз последњи конволутивни слој јернеља (64, 3, 3) и користи се последњи макс-пул сажимајући слој димензија (2,2). Након тога сигнал се исправља (енг. Flatten) у вектор колоне и пропагира кроз три потпуно повезана слоја, при чему сваки има, респективно, 128, 64 и 32 потпуно повезана вештачка неурона. Четврти потпуно повезани слој је промењив у зависности од броја излаза које очекујемо. Па тако четврти потпуно повезани слој прве мреже има 12, друге 16, а треће 15 потпуно повезана вештачка неурона. На излазу се појављују вредности које се поклапају са пољима у другим излазним датотекама апликације за означавање, описане у 3.1.2.4, изузев првог поља. Излазне вредности су у нормализованом опсегу [0, 1] и за прву мрежу има их 12, за другу 16 и за трећу 15.

На основу формула (2.3) и (2.4) можемо израчунати димензије сваког слоја, али и број параметара које је потребно да свака мрежа изучи:

Слој	Димензије јернеља	Димензије излаза	Параметри
Улаз	-	(100, 100, 1)	0
Конволутивни слој 1	(16, 3, 3)	(98, 98, 16)	160
<i>Max-pool</i> слој 1	(2,2)	(49, 49, 16)	0
Конволутивни слој 2	(16, 3, 3)	(47, 47, 32)	4640
<i>Max-pool</i> слој 2	(2,2)	(23, 23, 32)	0
Конволутивни слој 3	(16, 3, 3)	(21, 21, 64)	18496
<i>Max-pool</i> слој 3	(2,2)	(10, 10, 64)	0
<i>Flatten</i>	-	6400	0
Потпуно пов. слој 1	-	128	819328
Потпуно пов. слој 2	-	64	8256
Потпуно пов. слој 3	-	32	2080
Потпуно пов. слој 4	-	12 (16)* (15)**	396 (528)*(495)**

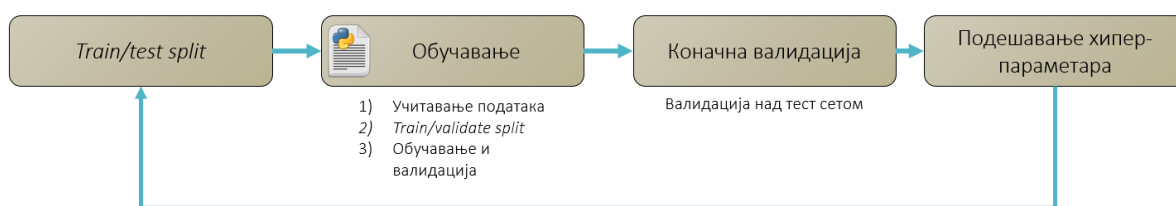
Табела 3.2 Димензије и параметри слојева првог \*(другог) \*\*(трећег) модела кориштене архитектуре неуронских мрежа

На основу података из табеле долазимо до укупног броја параметара који се могу изучити унутар сваке мреже. Првој је потребно да изучи 853 356 параметара, друга 853 488 параметара и трећа 853 455 параметара.

Да се још приметити, да су прва и друга мрежа обучавање да проналазе централне тачке ока, али на различитим фотографијама. Разлог из кога је одлучено да се те тачке од интереса преклапају између прве и друге мреже јесте да је циљ, осим решавања проблема, видети и показати способности неуронских мрежа и упоредити добијене резултате. Осврт на споменуто дат је у поглављу Евалуација.

### 3.2.2 Процес обучавања мрежа

Као што је поменуто у поглављу 2.3.5. процес обучавања неуронских мрежа је комплексан и итеративан процес. Према томе, скуп фотографија заједно са нормализованим излазним .csv датотекама из апликације за означавање који чине скуп за обучавање мрежа дели се на три дела: на део који служи као тест скуп након обучавања, на део којим се мрежа директно обучава и на део који служи за валидацију модела приликом обучавања. Овај процес још се назива и обучавање/валидација/тестирање подела скупа података (енг. *train/validate/test split*). На основу резултата над скупом за валидацију фино се подешавају хипер-параметри модела, док тест скуп служи за коначну валидацију модела и чине га подаци које мрежа никада није видела. Одређено правило по коме би се скуп података поделио на ова три дела не постоји и према томе тест скуп је величине 10 посто величине целокупног скупа, док се остатак дели на 80 посто за обучавање и 20 посто за валидацију модела.



Слика 3.7 Процес обучавања неуронске мреже

Методом насумичног одабира и праћењем метрике тачности валидације, изабрани су следећи хипер-параметри модела:

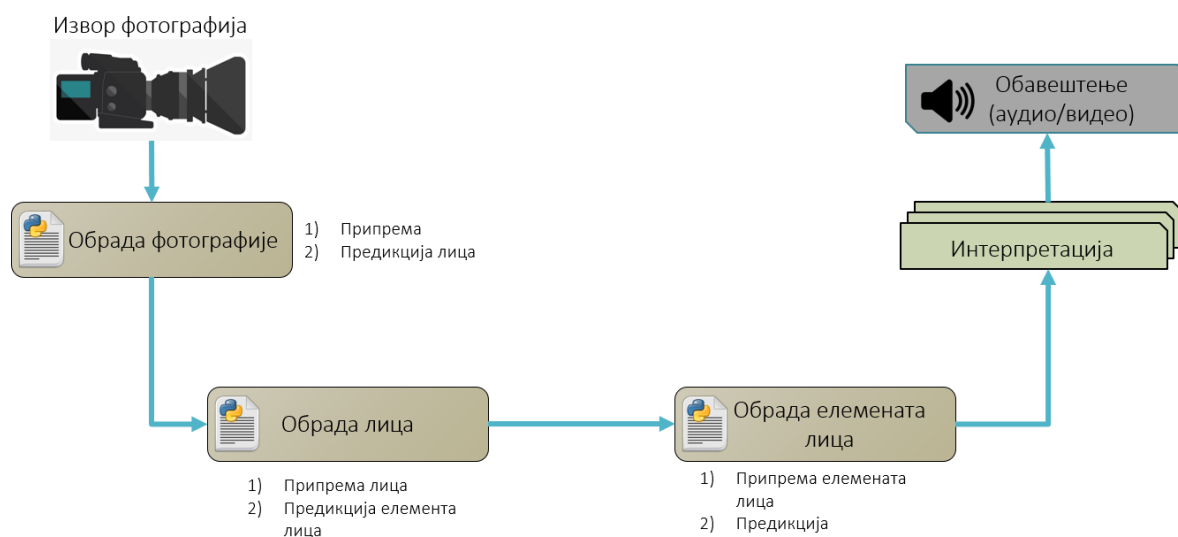
- Алгоритам оптимизације – *Adam*
- Функција губитака – *Mean squared error*

- Смањивање брзине обучавања – 0.5 са стрпљењем 15 епоха
- Величина групе података за обучавање – 2
- Фактор одбацивања неурона приликом обучавања – 0.25 после сваког макс-пул сажимајућег и потпуно повезаног слоја
- Број епоха – 350 са стрпљењем између 35 и 50 епоха

Показало се, на основу [21], али и током израде овог рада, да се најбољи резултати обучавања постижу уколико се користе алгоритми оптимизације са промењивим параметром брзине обучавања.

### 3.2.3 Структура апликације

Како смо претходно дефинисали архитектуру неуронских мрежа које ћемо користити у решавању проблема и обучили их, сада можемо дефинисати и структуру саме апликације за праћење пажње возача.



Слика 3.8 Структура апликације за праћење пажње возача

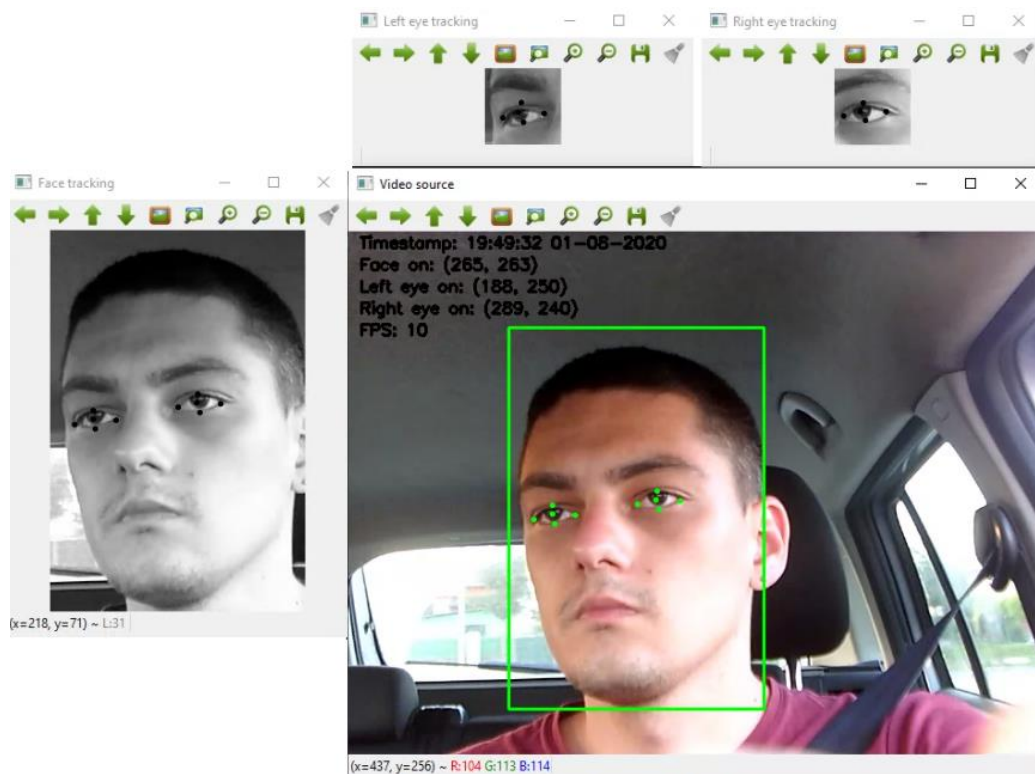
Идеја је да унутар возила постоји фиксиран извор фотографија који ће бити усмерен ка возачу и током вожње их прослеђивати на даљу обраду. Након што извор „ухвати“ фотографију, да би се проследила на неуронску мрежу, потребно је да се она припреми у одговарајући формат улаза који мрежа подржава. У нашем случају, извор фотографија је обична веб камера која снима фотографије у РГБ формату, резолуције 640x480. Таква фотографије се прво из РГБ простора боја претвара у сиву слику, односно из димензија 640x480x3 прелази у димензије 640x480x1. Након тога она се скалира на димензије 100x100 и нормализује се у опсег вредности [0, 1]. Таква је спремна за улаз

неуронске мреже и над њом се врши предикција. Ако мрежа пронађе лице, са такве, сиве фотографије уз помоћ денормализованих предикција центра лица врши се исецање фотографије лица, које се потом скалира на димензије 100x100, нормализује у опсег [0, 1] и прослеђује на другу мрежу која претражује елементе лица. Од свих елемената лица које друга мрежа може да прати (очи, нос, горњу и доњу усну) нама су битне очи. Уколико се на фотографији лица пронађе бар једно око, оно се уз помоћ денормализованих вредности предикција друге мреже исеца са слике лица и скалира на величину 100x100, нормализује у опсег [0, 1] и прослеђује трећој мрежи, како би се на основу њених, али и претходних предикција проценио тренутни степен пажње присутне код возача. Лева и десна рубна тачка, горња и доња рубна тачка и центар зенице су тачке од интереса које трећа мрежа прати и на основу њих доноси одлуку да ли је око отворено или затворено.

У току рада апликације, чувају се све вредности предикција и након временског интервала од 2.5 секунди проверава се у колико је обрађених фотографија лице окренуто у било коју страну и колико пута је детектовано да је било које око затворено. На основу тих вредности, ако је лице имало угао на више од 40 обрађених фотографија или ако је збир броја детектованих затворених очију већи од 35, сматра се да је код возача детектован низак ниво пажње.

Уколико се дође до закључка да је код возача присутан низак ниво пажње он се звучним сигналом обавештава и тиме се спречавају потенцијално непредвиђене и опасне ситуације.

Изглед финалне апликације може се видети на слици 3.9. Видео обавештење огледног је карактера и приказује денормализоване предикције мрежа у виду граничног правоугаоника који представља предикцију регије лица, тачака од интереса на очима, али и практичних информација о стању целокупног система. Да би се исцртала поменута регија и тачке од интереса, потребно је денормализовати излазне вредности лаким манипулисањем формуле (3.1).



Слика 3.9 Финална апликација за праћење пажње возача

Слика 3.9 приказује главни прозор и додатне прозоре апликације за праћење пажње возача. Главни прозор приказује све информације о систему и приказ целокупне обраде. Прозор који приказује лице приказује улаз у први модел неуронске мреже и резултате обраде који се шаљу на улаз другог модела. Прозори који приказују очи представљају улаз у трећи модел неуронске мреже и резултате њене обраде.

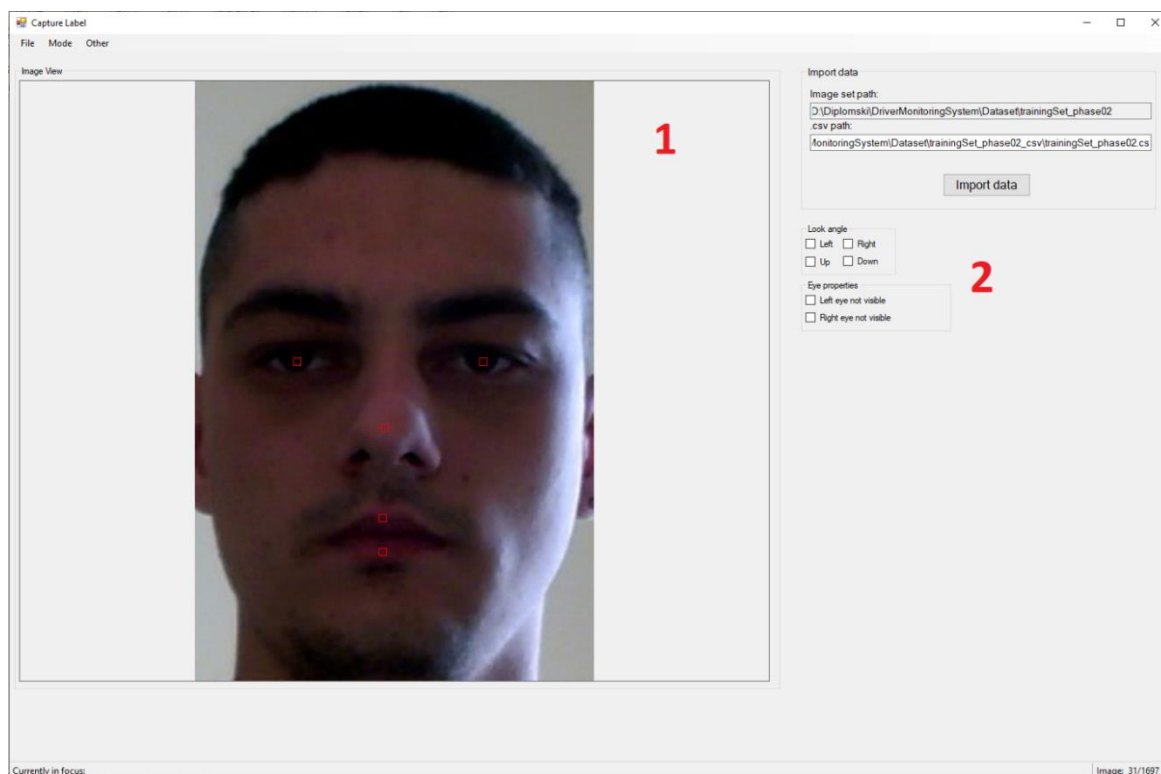


## 4 Програмско решење

У овом поглављу описана су програмска решења апликације за означавање података и апликације за праћење пажње возача.

### 4.1 Апликација за означавање података

Апликација за означавање података, као што је поменуто, реализована је као *Windows Forms Application (Microsoft .NET Framework 4.7.2)* уз помоћ *C#* програмског језика. Према томе, она се састоји од неколико битних *C#* класа које нам омогућавају да постигнемо жељене резултате, на што бржи и ефикаснији начин.



Слика 4.1 Апликација за означавање података

Ако погледамо слику 4.1, видимо да унутар простора за фотографије (енг. *Image view*) (1), постоје правоугаоници који служе за одређивање тачака од интереса. Ако претпоставимо да радимо у неком режиму рада, уз сваку слику се асоцира константан број правоугаоника, односно константан број координата на слици. Да би омогућили да се те координате чувају у меморији и да би оне биле лако доступне у сваком тренутку, уводимо две класе: *RectangleContainer* и *CoordinatesContainer<T>*.

Класа *RectangleContainer* нам омогућава да конструишемо објекат ове класе који садржи онолико правоугаоника, колико то захтева режим рада. Поља ове класе су листа правоугаоника и листа логички тачно/нетачно (енг. *True/False*) вредности које представљају информацију да ли је одређени правоугаоник фокусиран. Класа имплементира све потребне методе за брзо и ефикасно манипулисање правоугаоникима над једном сликом. Методе које она имплементира су: *getRectangles*, *getSize*, *setRectSize*, *getAllRectCoordinates*, *setRectCoordinates*, *setAllRectCoordinates*, *isInFocus*, *setFocus*, *contains*, *findInFocus*, *inFocusIndex*, *addToFocused*, *resetFocusList*, *addToRectSize*, *rescaleRect*, *resetState*.

Метода *getRectangles* као повратну вредност даје све правоугаонике из објекта типа ове класе у облику низа или листе, док метода *getSize* као повратну вредност носи

информацију о броју правоугаоника коју објекат садржи. *setRectSize* поставља правоугаоник одређеног индекса на произвољну величину, која је заједно са индексом прослеђена као параметар ове методе. *getAllRectCoordinates*, *setRectCoordinates* и *setAllRectCoordinates* служе, респективно, за добављање свих координата правоугаоника у виду листе целобројних вредности, за постављање координата правоугаоника одређеног индексом и постављање координата свих правоугаоника који су садржани у објекту типа ове класе уз помоћ листе целобројних вредности које представљају жељене координате, а прослеђују се као параметар функције. *isInFocus* врши проверу да ли је неки правоугаоник тренутно фокусиран, *setFocus* поставља фокус на правоугаоник неког индекса, *findInFocus* претражује листу правоугаоника и враћа онај правоугаоник који је у фокусу ако он постоји. *inFocusIndex* налази индекс фокусираног правоугаоника, *addToFocused* додаје на координате фокусираног правоугаоника целобројне вредности, *resetFocusList* доводи објекат у стање у коме нити један правоугаоник није фокусиран. *contains* проверава да ли се некаква тачка налази унутар било којег правоугаоника. *addToRectSize* правоугаонику жељеног индекса додаје или одузима целобројну вредност од његове величине, *rescaleRect* скалира правоугаоник одређеног индекса на константан омер ширине и висине уз произвољан фактор. *resetState* враћа објекат у иницијално стање, тако што ресетује листу фокусираних правоугаоника и поставља координате правоугаоника на жељене позиције.

*CoordinatesContainer<T>* јесте генеричка класа чије једино поље је приватно и типа *List<List<T>>*. Ова класа имплементира очување вредности координата правоугаоника за сваку учитану фотографију у апликацију за означавање. Замислимо то овако: уместо да у меморији имамо, за сваку фотографију, број правоугаоника који је одређен режимом рада, са свим својим атрибутима који нам нису потребни, ми ћемо за сваку слику очувати само оне информације које су нам од значаја, углавном координате правоугаоника. Тако ова класа имплементира неколико метода, које ће нам омогућити да што лакше приступамо тим координатама и вршимо потребну обраду. Методе које ова класа имплементира су: конструктор, конструктор са параметрима, конструктор копије, *addRow*, *replaceRow*, *getRow*, *getSize*, *setRowZero*, *getCoordinates*, *ConvertTo<U>*.

Метода *addRow* додаје нову врсту на зачеље листе са координатама у виду листе координата, *replaceRow* мења врсту у листи на жељеној позицији, врстом која јој се проследи. *getRow* враћа врсту координата на жељеном индексу. *getSize* враћа број врста, односно број елемената поља *List<List<T>>*. *setRowZero* поставља жељену врсту на

нулте вредности. *getCoordinates* повратна вредност јој је поље ове класе. *ConvertTo<U>* је метода која омогућава да се све вредности унутар свих врста конвертују у жељени тип (на програмеру је да буде обазрив приликом коришћења ове функције).

Како смо уз помоћ ове две класе омогућили очување свих вредности које су нам интересантне, сада имамо све елементе који су нам потребни да би се решио проблем означавања података. Класа *CaptureLabel* описује понашање свих елемената апликације за означавање података. Па тако неке од најбитнијих метода које ова класа имплементира су: *importButton\_Click*, *cleanUp*, *initMode*, *loadCoordinates*, *saveCoordinates*, *scrollUp*, *scrollDown*, *save*, *exportNormalized*.

Метода *importButton\_Click* представља улаз у сам програм и на основу свих информација подешених и унетих од стране корисника програм ће се иницијализовати у оном режиму који је тражен од њега. Такође, ово је функција из које се позивају функције за учитавање фотографија и учитавање информација из *.csv* датотеке, уколико је она прослеђена самој апликацији. *cleanUp* функција нам омогућава да несметано обављамо више учитавања у апликацију за означавање, без да морамо затварати и поново отварати апликацију за означавање. Она је доводи у стање пре првог учитавања. *initMode* иницијализује потребне променљиве и доводи апликацију у стање одабраног режима рада, конструише правоугаонике за означавање и поставља информације од значаја које се приписују одређеној фотографији, а одређене су режимом рада. Метода *loadCoordinates* омогућава учитавање координата правоугаоника за означавање и припрема их за приказ кориснику, а *saveCoordinates* похрањује координате правоугаоника за означавање. *scrollUp* и *scrollDown* су функције које заједно уз помоћ претходно две описане функције омогућавају кориснику да несметано пролази кроз фотографије унутар неког директоријума и да му се при томе приказују правоугаоници за означавање на позицијама које је већ означио или на позицијама које су стартне позиције пре него што корисник означава одређену фотографију. Метода *save* је метода уз помоћ које се позивају методе за конструисање и похрањивање *.csv* датотеке у коју ће бити уписане позиције у пикселима свих означених фотографија, а *exportNormalized* омогућава да се подаци припреме и позову методе које конструишу и похрањују преостале две *.csv* датотеке које чине излаз из апликације за означавање, а у које су уписане нормализоване вредности координата у пикселима и вредности потребне за денормализацију нормализованих вредности.

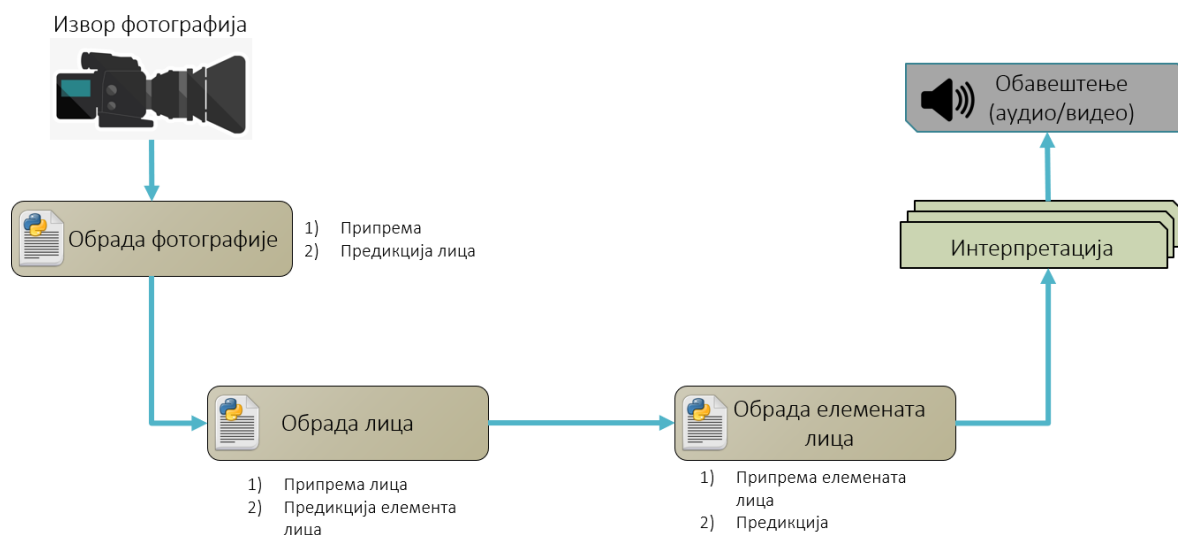
Статичка класа *Utilities* имплементира све функционалности које су хијерархијски ниже целине у односу на функционалности имплементирани у класи *CaptureLabel*, и чине ситније целине које се позивају више пута током једне сесије апликације за означавање података. Па тако имамо методе као што су: *parseImagesToList*, *writeToCSV*, *writeMinMax*, *parseCSV*, *normalizeOutput*, *correctFaceCoordinates*.

Метода *parseImagesToList* омогућава да се из директоријума који је прослеђен апликацији за означавање ишчитају путање до фотографија подржаног формата и врате као повратна вредност ове методе. *writeToCSV* је генеричка метода, која омогућава конструисање *.csv* датотеке на основу свих потребних података који су јој прослеђени, а чији тип може да буде променљив у зависности од тога да ли се конструише *.csv* датотека са вредностима у пикселима или нормализованим вредностима. *writeMinMax* такође је генеричка метода којом се конструише конструисање *.csv* датотека у коју ће бити уписане вредности потребне за нормализацију и денормализацију података. Уз помоћ *parseCSV* методе учитавају се координате правоугаоника из претходно конструисане *.csv* датотеке од стране апликације за означавање, а која јој је прослеђена као улаз. *normalizeOutput* служи са нормализовање вредности свих означених правоугаоника, а позива се на захтев корисника да се вредности упишу у нормализовану *.csv* датотеку. Пошто се у меморији чувају координате горње леве тачке сваког правоугаоника, а у *Face mode* режиму рада је потребно имати средишњу тачку лица, користимо методу *correctFaceCoordinates* која има способност да конвертује те координате једне у друге и обратно.

Као део апликације за означавање налази се и статичка класа *Constants* у којој су похрањене све константе од значаја, као што су стартне позиције правоугаоника за означавање када нису учитане вредности из претходно конструисаних *.csv* датотека, константне вредности тастера којима се фокусирају различити правоугаоници путем тастатуре, итд.

## 4.2 Апликација за праћење пажње возача

Апликација за праћење пажње возача имплементирана је уз помоћ *Anaconda* 4.8.2 дистрибуције отвореног кода *Python* 3.7 програмског језика, која се често користи за развој апликација машинског учења. Такође, користе се и друге библиотеке које су од значаја за развој таквих апликација: *Tensorflow* 2.1, *numpy*, *pandas*, *OpenCV*, *pillow*.



Слика 4.2 Апликација за праћење пажње возача

Пре него што апликација за праћење пажње возача уђе у петљу којој се фотографије обрађују потребно је да се у њу саму учитају обучени модели неуронских мрежа. Након тога када извор фотографија проследи фотографију која је на реду за обраду, она пролази кроз неколико фаза пре него што се она класифицира као фотографија на којој је код возача присутан одређени степен пажње:

1. **Обрада фотографије** – прослеђена фотографија се припрема за улазе неуронске мреже уз помоћ *OpenCV* библиотеке, која имплементира функцију уз чију помоћ фотографија прелази из РГБ простора боја у сиву слику. Таква црно-бела фотографија се уз помоћ функције *resizeAndNormalizeImage* скалира на димензије 100x100, а вредност сваког пиксела се нормализује у опсег вредности [0, 1]. Над таквом скалираном и нормализованом фотографијом ради се предикција уз помоћ првог модела неуронске мреже и ако се установи да се на слици налази лице, обрада се преноси у следећу фазу. У супротном, поступак се понавља све док са извора фотографије не пристигне фотографија на којој неуронска мрежа успе да пронађе лице.

2. **Обрада лица** – уколико је у претходној фази модел прве неуронске мреже пронашао лице, са сиве фотографије оригиналних димензија исеца се део са лицем уз помоћ денормализованих предикција првог модела. Ово је обезбеђено кроз имплементацију функције *cropFace* којој се прослеђује оригинална црно-бела фотографија, али и предикције првог модела. Унутар ње денормализују се вредности ширине и централних координата лица и врши се исецање. Она као повратну вредност враћа фотографију лица. По узору на претходну фазу, уз помоћ функције *resizeAndNormalizeImage*, лице се скалира на димензије 100x100, а вредност сваког пиксела се нормализује у опсег вредности [0, 1]. Тек тада се могу вршити предикције уз помоћ модела друге неуронске мреже. Уколико модел друге неуронске мреже врати предикције, које одговарају на питање да ли на фотографији лица постоје очи, да бар једно око постоји, прелази се у следећу фазу. У супротном интерпретација података се ослања само на предикције првог модела.
3. **Обрада елемената лица** – сада када знамо да се на фотографији лица налази бар једно око, уз помоћ имплементиране функције *cropEyes* врши се припрема фотографија очију по узору на претходну фазу. Ова функција ће уз помоћ денормализованих предикција другог модела неуронске мреже, са фотографије лица исећи делове на којима се претпоставља да садрже очи и вратити их као своју повратну вредност. Сада се те фотографије очију, такође, по узору на претходне две фазе, скалирају на димензије 100x100, а вредност сваког пиксела се нормализује у опсег вредности [0, 1] уз помоћ функције *resizeAndNormalizeImage*. Модел треће неуронске мреже сада има све што му је потребно да врши предикције и одреди да ли је око отворено или затворено.
4. **Интерпретација** – током рада апликације, унутар временског интервала од 2.5 секунде, сакупљају се све предикције и уз помоћ функције *checkAttention*, која се извршава у посебној нити, процењује се да ли је код возача присутно одсуство пажње. Таква процена врши се на основу вредности поља првог модела која се односе на то да ли је лице под углом и поља из трећег модела која се односе да ли су очи отворене. Уколико се у том временском интервалу пронађе 40 предикција које указују да лице има било који угао или таквих да је унутар 35 обрађених фотографија бар једно око затворено, сматра се да је возачев ниво пажње опао.
5. **Обавештење** – о стању система корисник се обавештава у два вида:

- a. *Аудио обавештење* – аудио обавештење представља примаран вид обавештења унутар самог возила и врши се праводобно на основу интерпретираних података.
- b. *Видео обавештење* – видео обавештење је демонстративног карактера и служи нам како би боље разумели понашање целокупног система кроз све фазе обраде. На оригиналној РГБ фотографији, која је прослеђена на обраду од стране извора фотографија, исцртава се гранични правоугаоник који је одређен предикцијама првог модела и означава део у коме се према тим предикцијама налази лице. Унутар њега налази се и по пет тачака које представљају четири рубне тачке ока и центар зенице, а које се исцртавају на основу денормализованих вредности модела треће мреже. Уколико је дошло до спознаје да су очи затворене, те тачке мењају боју из зелене у црвену. Ову функционалност имплементира функција *drawPredictionOnImage*. Такође, позива се и функција *showInfo*, која на оригиналну фотографију исписује корисне информације као што су: просечан број обрађених фотографија у секунди, денормализоване предикције координата центра лица, левог и десног ока.

Како предикције неуронских мрежа знају бити нестабилне, односно како се у току неког временског интервала зна појавити мањи број предикција које знатно одскачу од резултата који се интуитивно очекују, након предикција мрежа користи се техника која нам омогућава „пеглање“ оваквих нежељених резултата, али уноси кашњење у систем. Принцип на коме се заснива јесте да се унутар одређеног прозора предикција константне величине рачуна средња вредност прозора. Ова техника се на енглеском назива *Simple Moving Average*. Поступак је итеративан и прозор се помера у складу са пристизањем нових предикција ( $A_n$ ), тако да се средња вредност рачуна над  $n$  најмлађих предикција.

$$SMA = \frac{A_1 + A_2 + \dots + A_n}{n}, \quad (4.1)$$

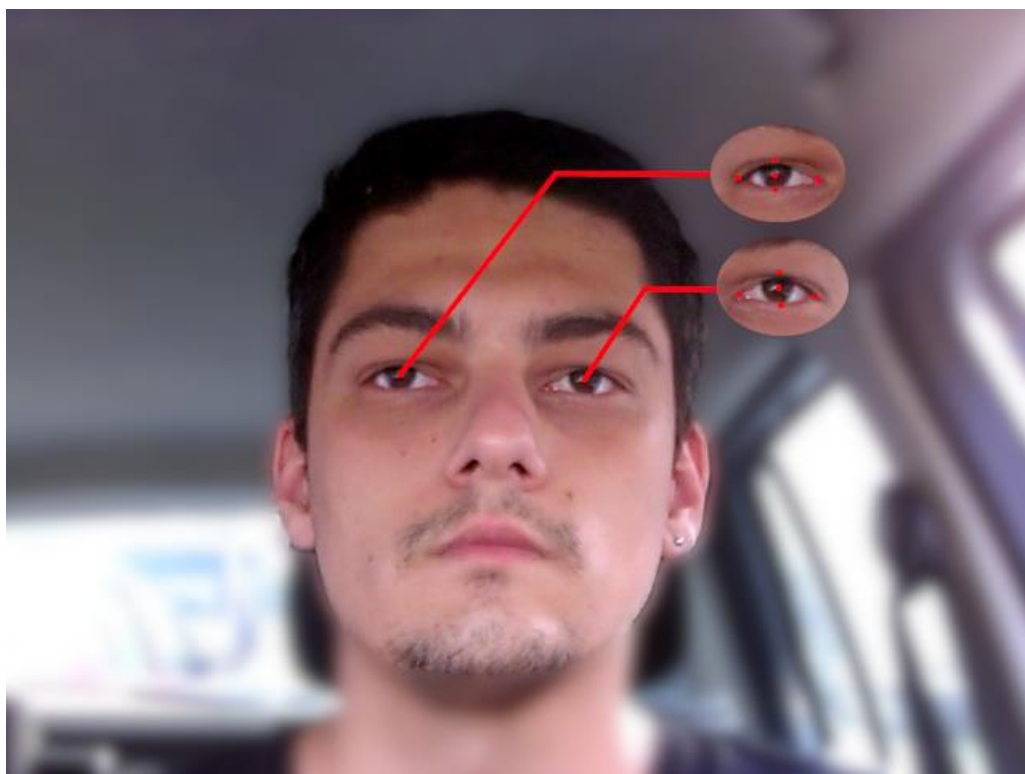


### 4.3 Концепт модела фокусиране пажње

Фокусирана пажња представља когнитивну могућност мозга да се усмери на детаље, односно на некакав стимулус током неког временског периода. Као таква, та способност се разликује од човека до човека и уколико је она на већем нивоу то је и човекова способност ефикасног обављања разноразних свакодневних активности већа. Као пример фокусиране пажње можемо узети студента који седи у читаоници, учи и занемарује све остале факторе који му се природно намећу. Његова усмереност ка градиву и неусмереност ка свему осталом јесте фокусирана пажња. Другим речима његовом стимулус током времена представља градиво, док све остало су супресовани стимулуси који нису од значаја. Једна од одлика, али и мана фокусиране пажње јесте да она може, адекватним обучавањем когнитивне стимулације, да се побољша, али исто тако, узроковано траумом или болешћу да се погорша. [22], [23], [24]

Модели фокусиране пажње нису стран појам, како у пољу рачунарске визије, тако и у неким другим пољима и за пример тога може се пронаћи модел за препознавање аутомобилских таблица. [25]

Интересантно је да ако се програмско решење апликације за праћење пажње возача посматра као црна кутија од које тражимо да на основу улазне фотографије на којој се налази мноштво информација процени степен пажње неког лица, имамо модел фокусиране пажње. Његове стимулусе, пре свега, представљају они делови фотографије на којима се налази лице, чиме он своју пажњу усмерује на те регије, док све остало одбацује. Надаље, он своју пажњу усмерава на све ситније детаље, као што су очи и детаљи на њима и заједно уз атрибуте као што су: у којем правцу је лице окренуто, у ком правцу је усмерен поглед, да ли су очи отворене или затворене; даје процену пажње. Другим речима, на основу датог контекста, модел је способан да да процену чији је ниво комплексности еквивалентан комплексности процене човековог мозга. Такође, анализирањем резултата тог модела и обучавањем по целинама, значајно се могу унапредити његове способности. Визуализација идеје приказана је на слици 4.3.



Слика 4.3 Концепт модела фокусиране пажње

## 5 Евалуација

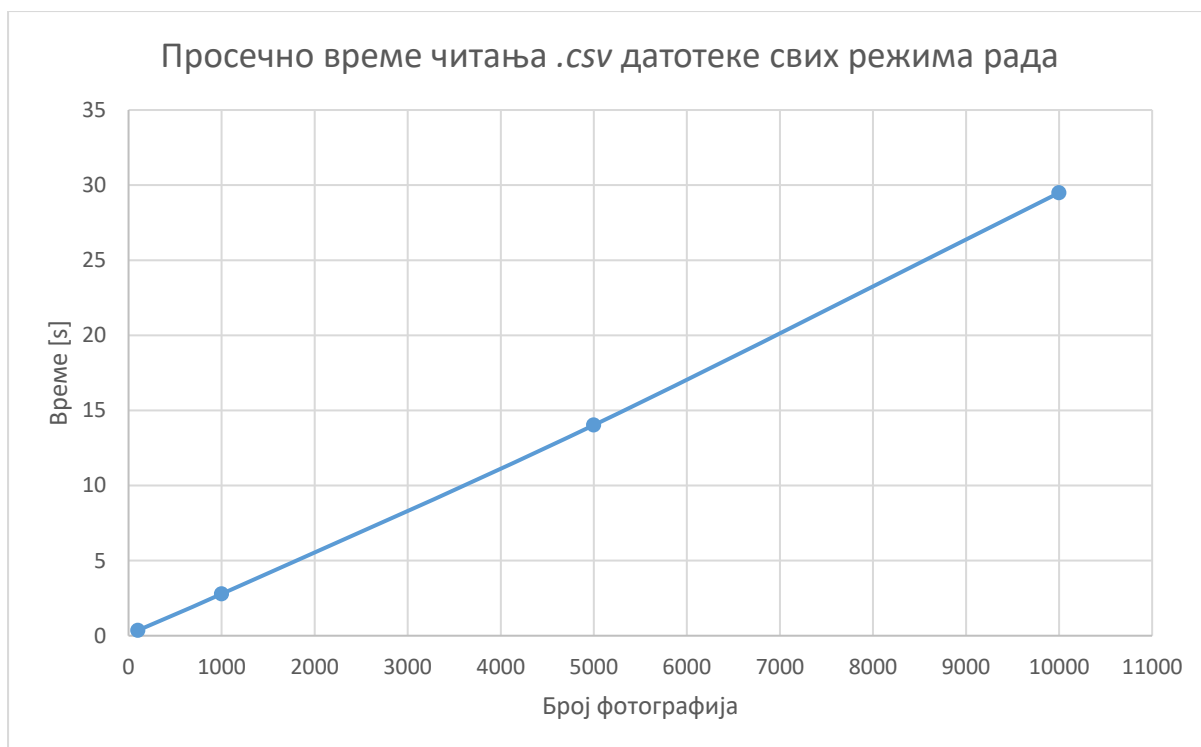
У овом поглављу дат је осврт на постигнуте резултате апликације за означавање и апликације за праћење пажње возача.

### 5.1 Евалуација апликације за означавање података

Као што је споменуто у 3.1.2. апликација за означавање података током свог рада не чува стварне вредности координата већ их прерачунава онда када се од ње затражи уписивање у *.csv* датотеку или када се таква датотека учитава (како би се наставио прекинути процес означавања). С обзиром на ово и на карактеристику да је у тренутку означавања одређене фотографије само она учитана у апликацију, критични кандидати за њену евалуацију су време потребно да се прочитају подаци из *.csv* датотеке, време потребно за упис и конструкцију *.csv* датотека и потрошња радне меморије.

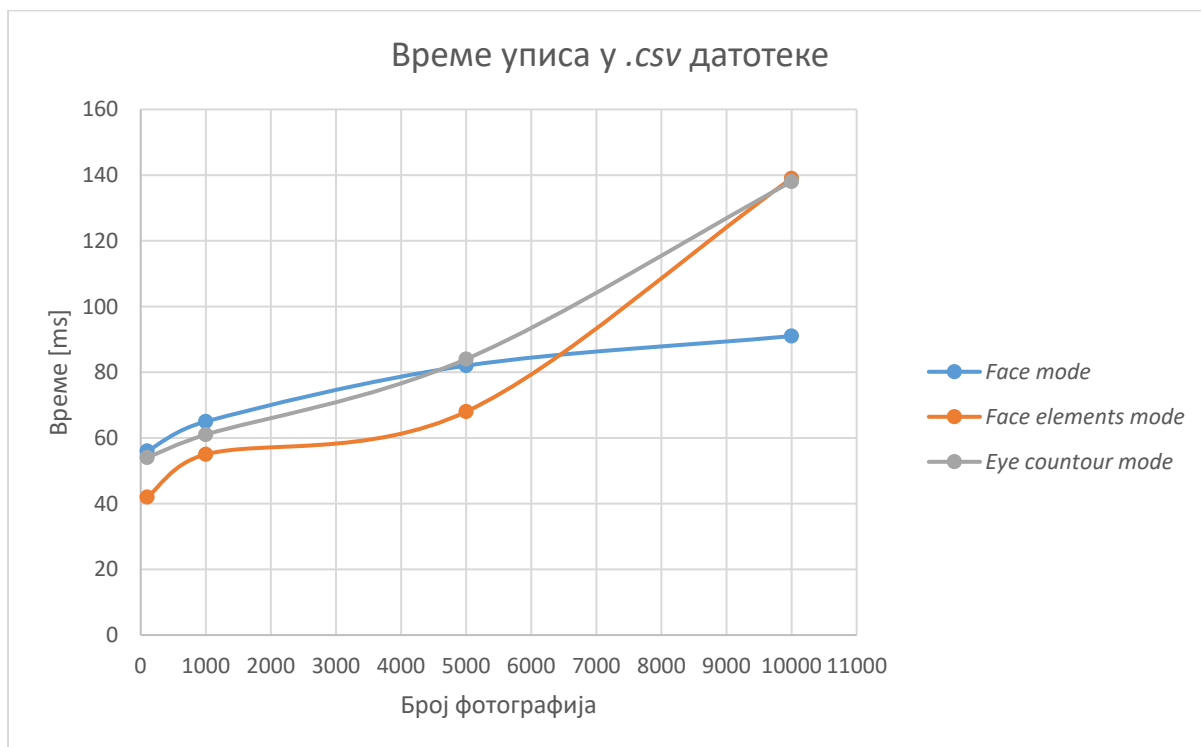
Сва мерења су обављена приликом првог покретања апликације уз помоћ *Stopwatch* класе која је препоручена за мерење времена извршавања делова кода. Испитивања су се вршила на *Intel i7-6700hq* процесору са 16ГБ радне меморије.

Графикон 5.1 приказује време потребно за читање координата из *.csv* датотеке до тренутка када је апликација спремна за рад; за 100, 1 000, 5 000 и 10 000 означених фотографија.

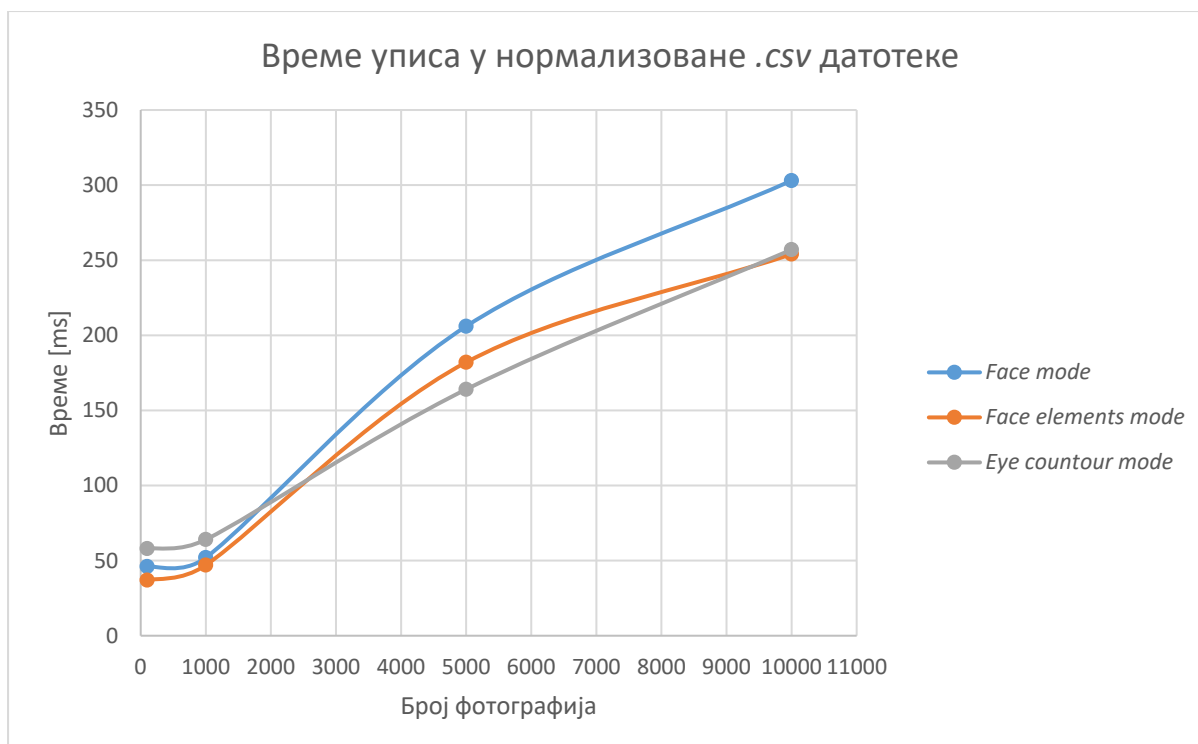


Графикон 5.1 Просечно време читања .csv датотеке свих режима рада

Графикони 5.2 и 5.3 приказују време потребно за конструкцију и упис координата у .csv датотеке за 100, 1 000, 5 000 и 10 000 означених фотографија.



Графикон 5.2 Време уписа у .csv датотеке



Графикон 5.3 Време уписа у нормализоване .csv датотеке

Потрошња радне меморије апликације за означавање података, када у њу није учитан нити један податак је 17МБ, а када се у њу учита највећа .csv датотека, која чува информације за око 10 000 фотографија износи 29МБ.

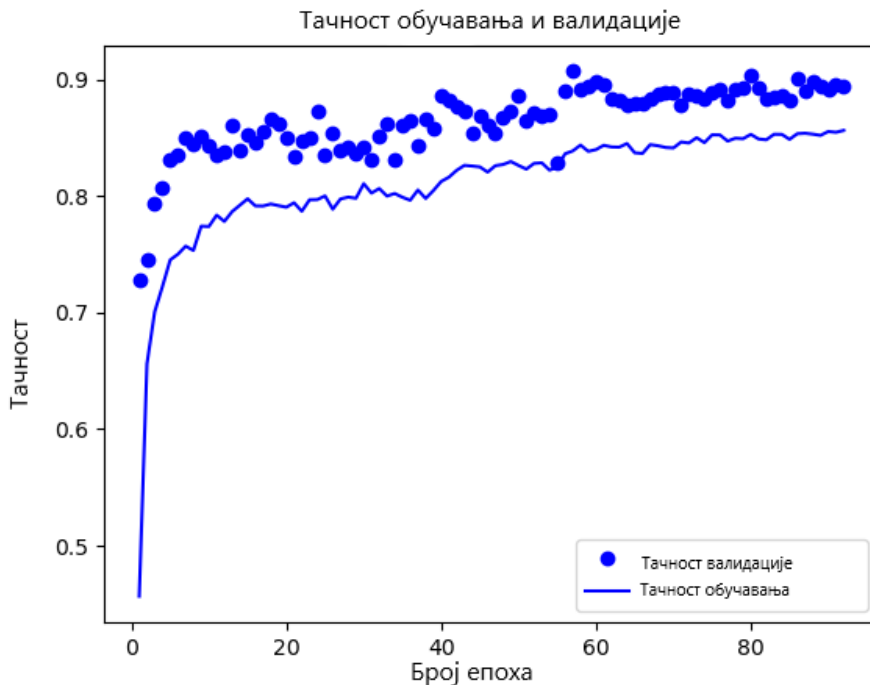
## 5.2 Евалуација апликације за праћење пажње возача

У овом поглављу дат је осврт на резултате апликације за праћење пажње возача пропраћене графиконима који представљају резултате обучавања неуронских мрежа на основу очекиваних означених података и предикција над тест скупом. Дат је осврт на временску потрошњу појединих делова програмског кода и одговор на питање да ли је било потребно уводити три неуронске мреже.

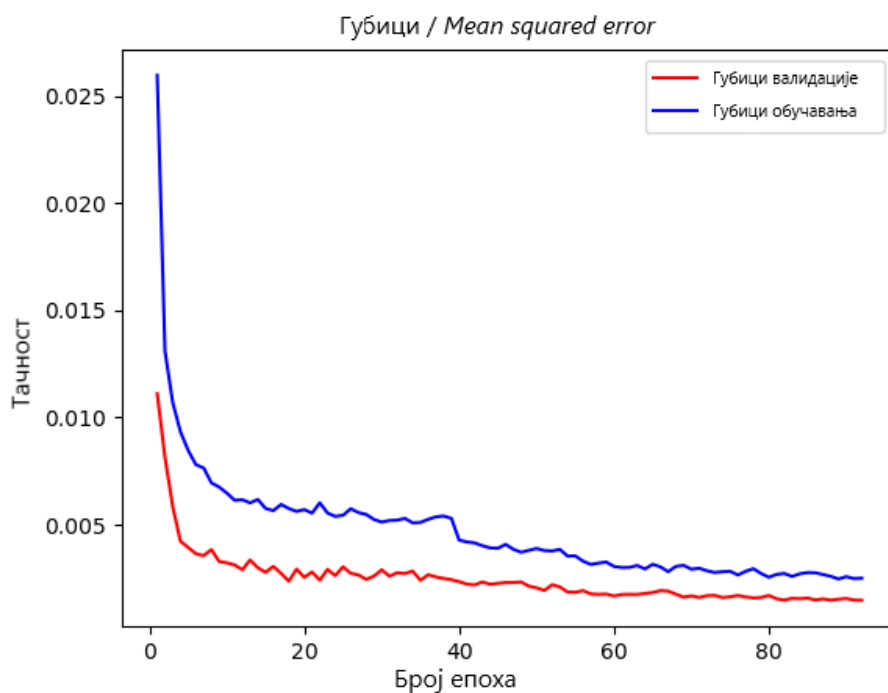
Ради смањења опширности рада издвојени су најзанимљивији резултати обучавања.

### 5.2.1 Евалуација првог модела неуронске мреже

На графиконима 5.4 и 5.5 приказани су тачности обучавања и валидације модела, као и губици током процеса обучавања.



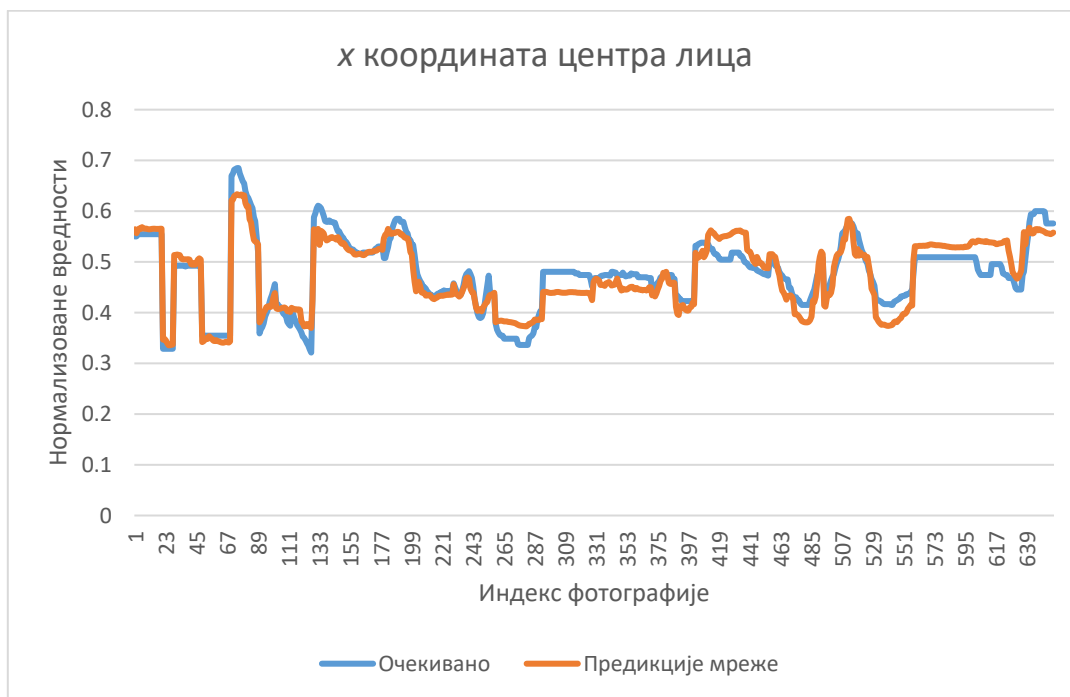
Графикон 5.4 Тачност обучавања и валидације првог модела



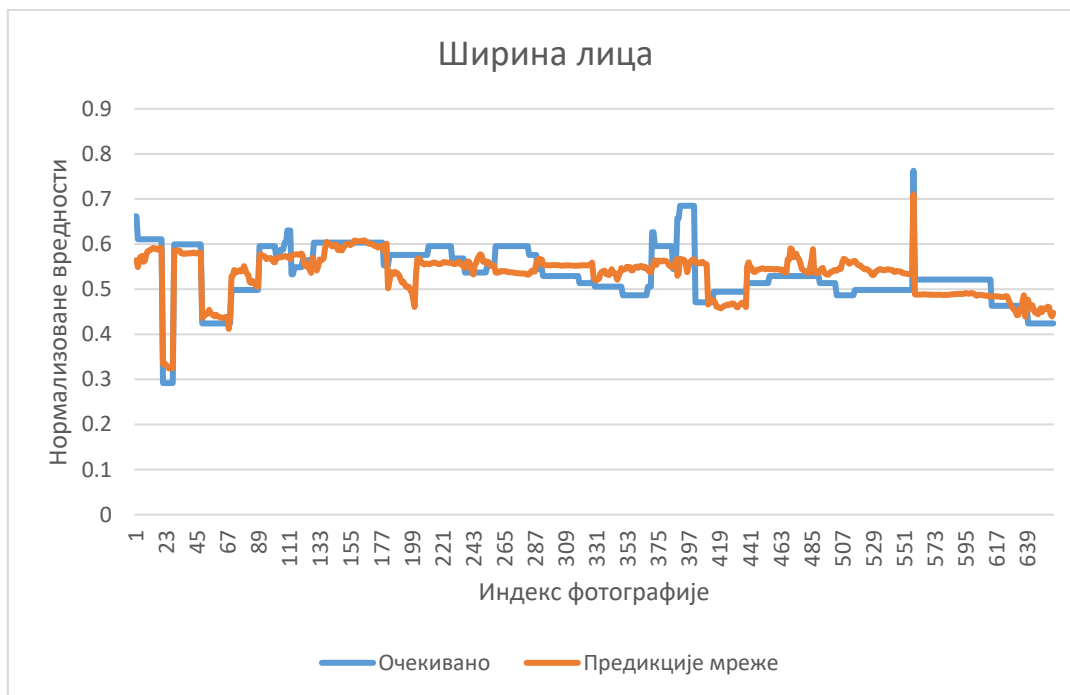
Графикон 5.5 Губици првог модела

Са графикана 5.4 може се видети да постигнута тачност валидације износи 0.8949.

У наставку су дати неки случајеви поређења очекиваних и добијених резултата над тест скупом.



Графикон 5.6 x координата центра лица



Графикон 5.7 Ширина лица

На основу графикана 5.6 и 5.7 може се видети са коликом прецизношћу први модел проналази центар лица по  $x$  координати и ширину лица над тест скупом података које није видела приликом обучавања. Разлика у предикцијама мреже у односу на означене податке у неким случајевима, на основу графикана износи до  $\sim 10\%$ . Код интерпретације ових вредности треба водити рачуна о људској процени и грешци приликом означавања. Приликом означавања даје се оквирна процена где се налази центар лица, али и информација о његовој ширини, према томе није искључена могућност да је неуронска мрежа заправо дала бољу процену него што је то урадио човек. Чак и ако то није тако, проблем за даљу обраду не представља и ако се исече пар пиксела више или мање.

Параметар	Тачност [%]
<i>noFace</i>	100
<i>Face x</i> координата	91.19
<i>Face y</i> координата	98.72
<i>Left eye x</i> координата	73.64
<i>Left eye y</i> координата	84.89
<i>Right eye x</i> координата	79.18
<i>Right eye y</i> координата	85.80
<i>Left</i>	97.42
<i>Right</i>	95.74
<i>Up</i>	100
<i>Down</i>	100
<i>Face width</i>	82.98

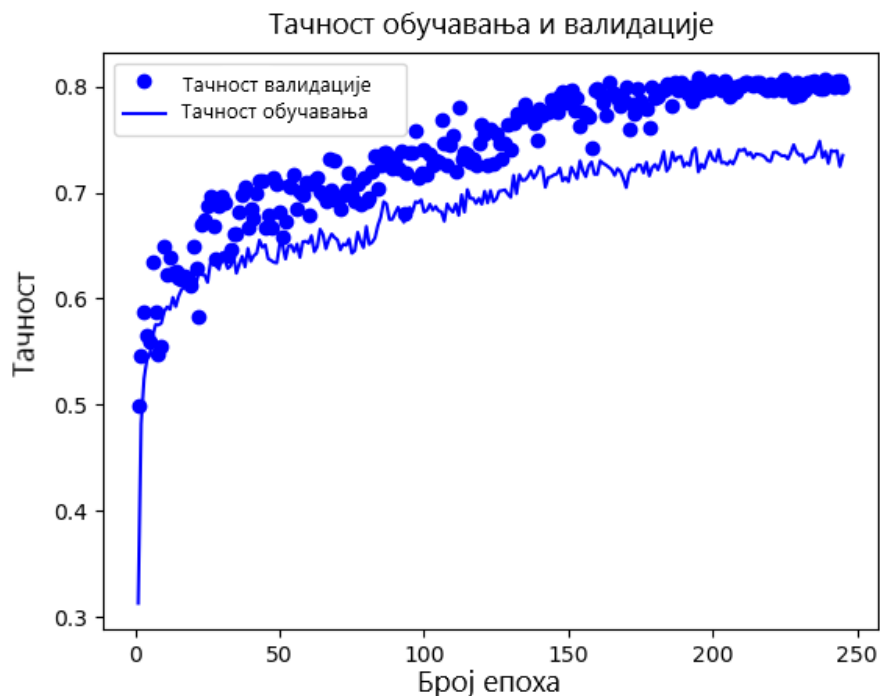
Табела 5.1 Проценат тачних предикција над тест сетом првог модела

У табели 5.1 могу се видети тачности предикција свих параметара првог модела мреже над тест сетом података. Предикцију сматрамо тачном уколико је процентуална разлика између очекиване вредности и вредности предикције мања од  $10\%$  када је реч о параметрима који представљају координате. У осталим случајевима сматрамо да је предикција тачна уколико вредност предикције прелази праг провођења  $0.5$  када је очекивана вредност једнака  $1$ , а мања од тог прага уколико је очекивана вредност једнака  $0$ .

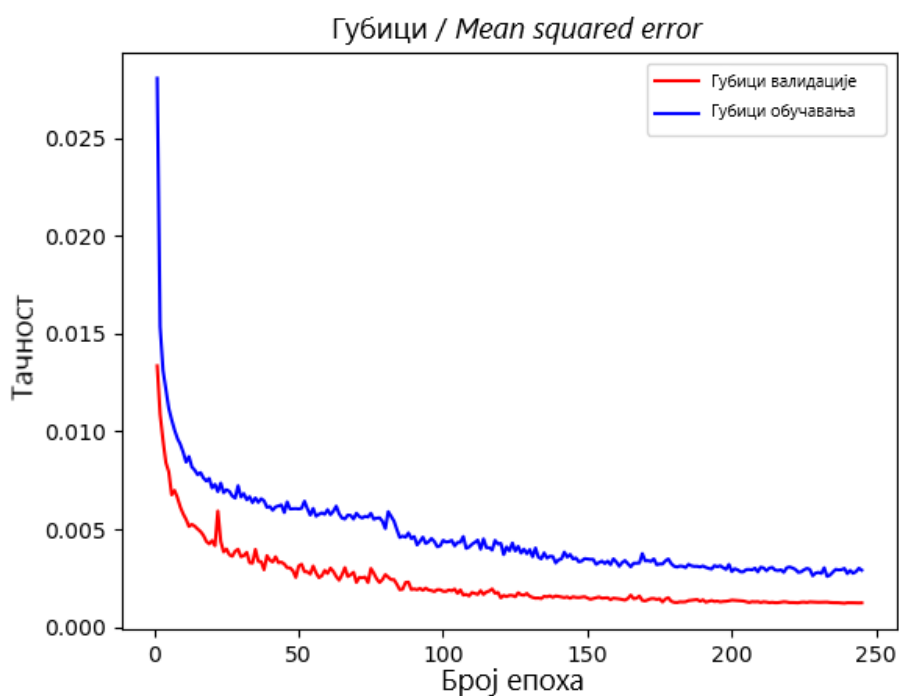


### 5.2.2 Евалуација другог модела неуронске мреже

На графиконима 5.8 и 5.9 приказани су тачности обучавања и валидације модела, као и губици током процеса обучавања.



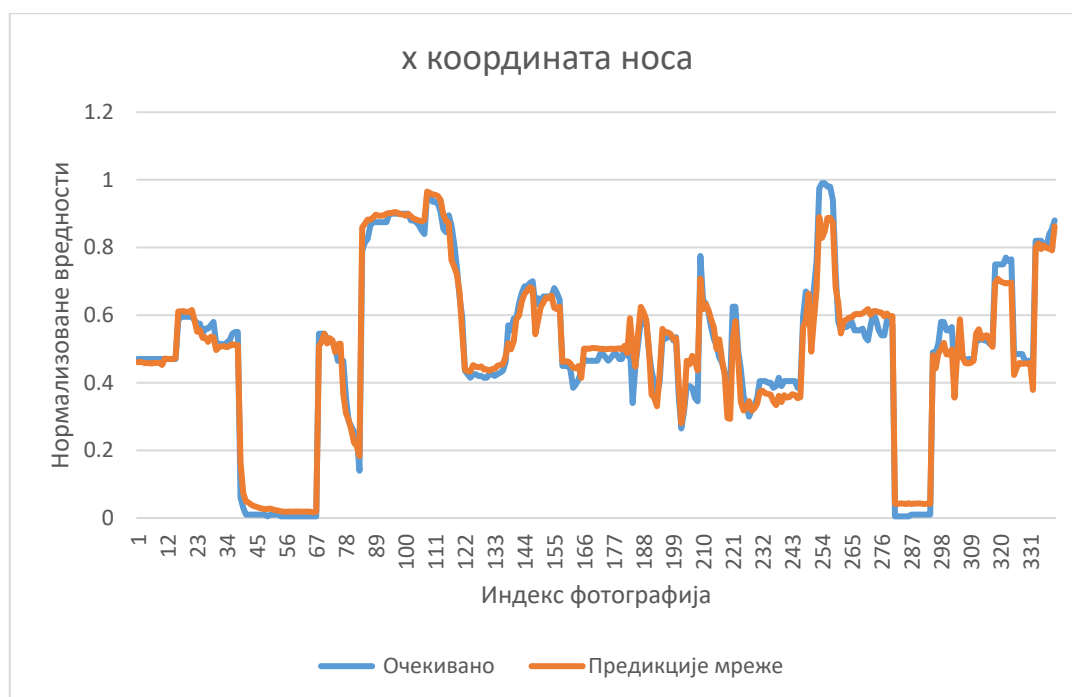
Графикон 5.8 Тачност обучавања и валидације другог модела



Графикон 5.9 Губици другог модела

Са графикана 5.8 може се видети да је постигнута тачност валидације око 0.8, прецизније 0.80759. Експериментално, а и поређењем очекиваних и добијених резултата над тест скупом, је утврђено да се уклањањем вредности унутар скупа за обучавање које носе информацију да ли је поглед усмерен горе, доле, лево или десно постижу много боље тачности  $\sim 0.9$ . На основу тога, долазимо до закључка да би скуп за обучавање могао бити бољи у смислу проширивања са случајевима у којима доминирају такве вредности и њихове комбинације.

У наставку су дати неки случајеви поређења очекиваних и добијених резултата над тест скупом.



Графикон 5.10 x координата носа

На графикону 5.10 може се видети са колико малом грешком мрежа одређује x координату носа на тест скупом података. С обзиром да је тест скуп део скупа података који мрежа никада није видела, може се рећи да је мрежа успела јако добро да генерализује специфичности сигнала и обезбеди поуздане резултате. Наравно, како нити један реалан систем није идеалан, у некаким екстремним случајевима предикције мреже могу знатно да одскачу од жељених резултата.



Графикон 5.11 Модел 2 - поглед усмерен лево

Вертикална оса графикона 5.11 представља вероватноћу да је поглед возача усмерен у лево. У случају да је та вероватноћа већа од 0.5 сматрамо да је поглед усмерен у том правцу. Као што се види, у неким случајевима резултати мреже одскачу од очекиваних резултата и ово би био добар пример када неуронска мрежа даје лажно позитивне резултате, такође може се приметити да она у одређеном распону индекса фотографија није успела да детектује да је поглед усмерен у лево. Решење оба проблема било би проширивање скупа података са оваквим случајевима.

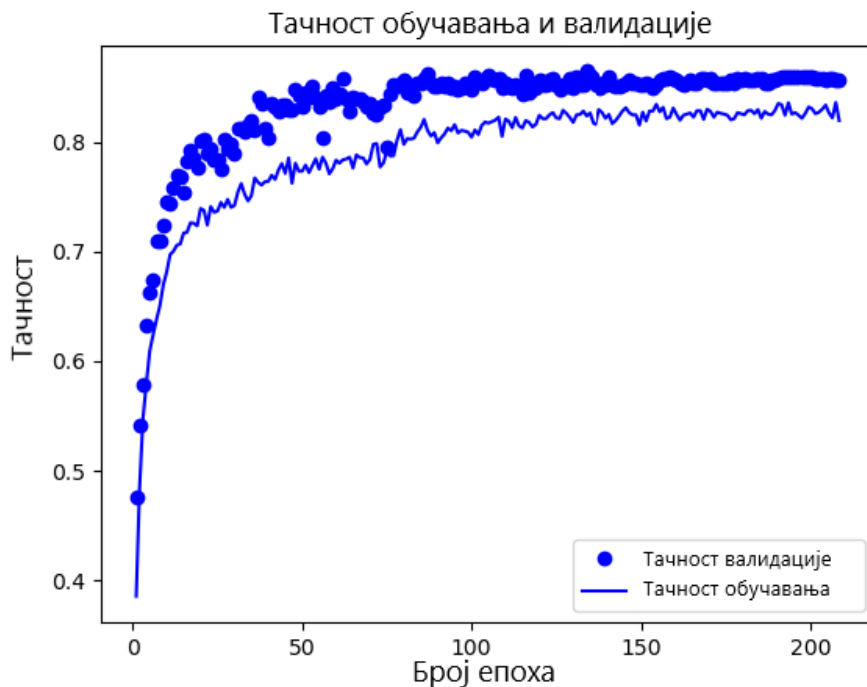
Параметар	Тачност [%]
<i>noLeftEye</i>	98.82
<i>noRighEye</i>	98.82
<i>Left Eye x</i> координата	52.35
<i>Left Eye y</i> координата	80.59
<i>Right Eye x</i> координата	69.41
<i>Right Eye y</i> координата	77.35
<i>Nose x</i> координата	70.59
<i>Nose y</i> координата	78.53
<i>Mouth up x</i> координата	79.71
<i>Mouth up y</i> координата	74.12
<i>Mouth down x</i> координата	70.00
<i>Mouth down y</i> координата	59.12
<i>Left</i>	94.41
<i>Right</i>	96.47
<i>Up</i>	100
<i>Down</i>	99.12

Табела 5.2 Проценат тачних предикција над тест сетом другог модела

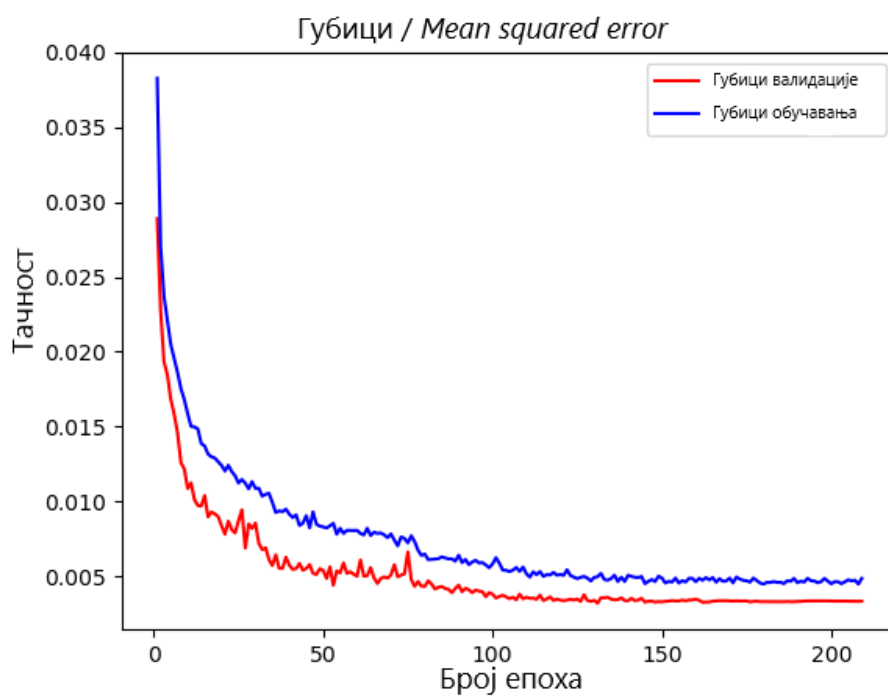
У табели 5.2 могу се видети тачности предикција свих параметара другог модела мреже над тест сетом података. Предикцију сматрамо тачном уколико је процентуална разлика између очекиване вредности и вредности предикције мања од 10% када је реч о параметрима који представљају координате. У осталим случајевима сматрамо да је предикција тачна уколико вредност предикције прелази праг провођења 0.5 када је очекивана вредност једнака 1, а мања од тог прага уколико је очекивана вредност једнака 0.

### 5.2.3 Евалуација трећег модела неуронске мреже

На графиконима 5.12 и 5.13 приказани су тачности обучавања и валидације модела, као и губици током процеса обучавања.



Графикон 5.12 Тачност обучавања и валидације трећег модела

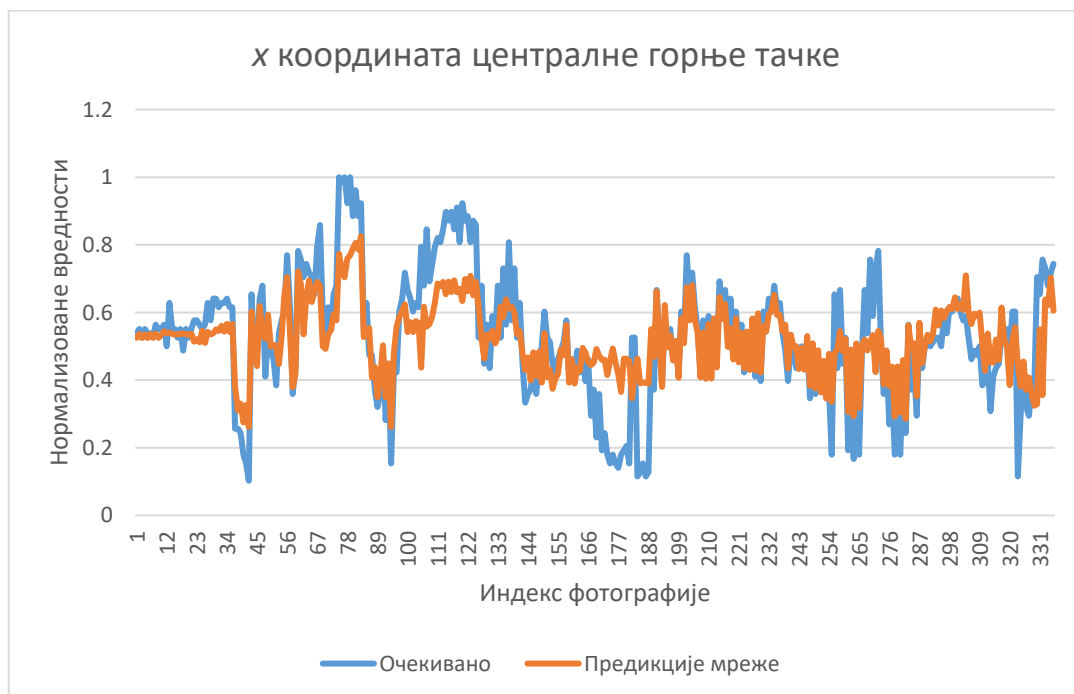


Графикон 5.13 Губици трећег модела

Са графикана 5.12 може се видети да је постигнута тачност валидације 0.86426. У пракси се показало да, за потребе овог рада, ова тачност даје задовољавајуће резултате.



Графикон 5.14 у координата центра зенице



Графикон 5.15 x координата централне горње тачке ока

Графикони 5.14 и 5.15 фино приказују како се мрежа понаша када је тест скуп прошаран са екстремним очекиваним вредностима. Како до тих екстремних случајева

долази када други модел направи изразито лошу предикцију центра ока, овакви резултати су прихватљиви. Упркос томе, ваљало би у скуп за обучавање додати и таквих примера.

Параметар	Тачност [%]
<i>eyeClosed</i>	92.26
<i>Center up x</i> координата	61.01
<i>Center up y</i> координата	70.54
<i>Center x</i> координата	65.67
<i>Center y</i> координата	77.08
<i>Center down x</i> координата	55.36
<i>Center down y</i> координата	76.95
<i>Left point x</i> координата	39.04
<i>Left point y</i> координата	64.88
<i>Right point x</i> координата	51.19
<i>Right point y</i> координата	73.51
<i>Left</i>	80.06
<i>Right</i>	96.13
<i>Up</i>	97.02
<i>Down</i>	88.99

Табела 5.3 Проценат тачних предикција над тест сетом трећег модела

У табели 5.3 могу се видети тачности предикција свих параметара трећег модела мреже над тест сетом података. Предикцију сматрамо тачном уколико је процентуална разлика између очекиване вредности и вредности предикције мања од 15% када је реч о параметрима који представљају координате. У осталим случајевима сматрамо да је предикција тачна уколико вредност предикције прелази праг провођења 0.5 када је очекивана вредност једнака 1, а мања од тог прага уколико је очекивана вредност једнака 0.

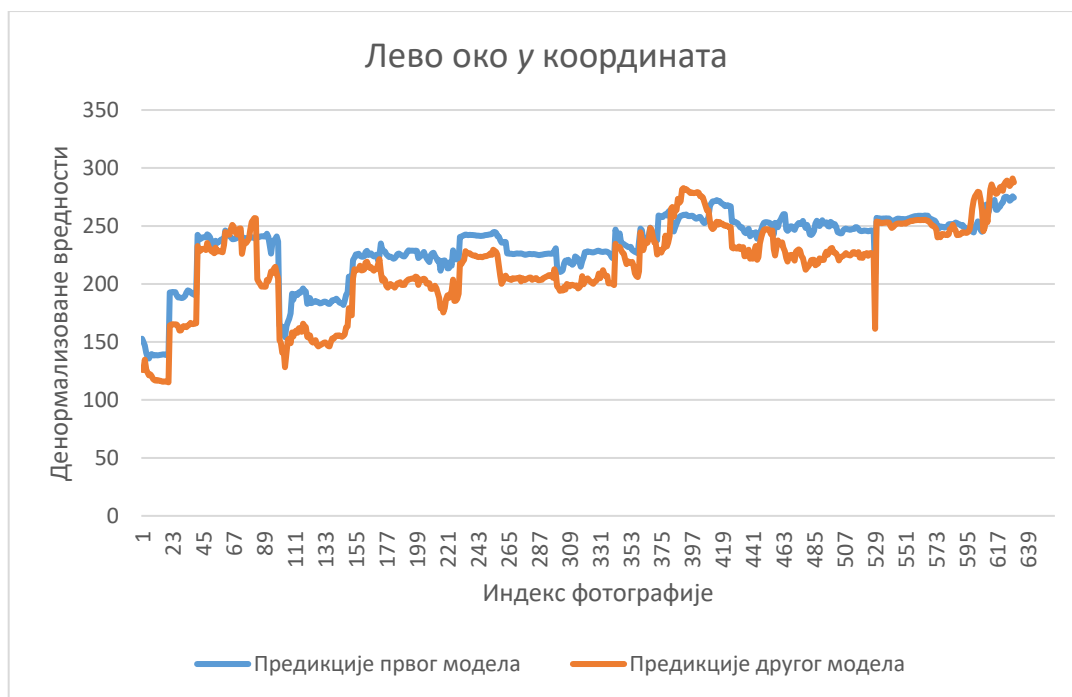
### 5.2.4 Могућност искључивања другог модела

Раније је поменуто да су први и други модел неуронских мрежа обучени да претражују простор фотографије и проналазе исте елементе, централне тачке очију. У овом поглављу, на основу добијених резултата, разматра се могућност искључивања другог модела, односно врши се испитивање да ли очи представљају сувише ситне елементе унутар целе фотографије да би их први модел неуронске мреже пронашао са задовољавајућом прецизношћу.



Графикон 5.16 Предикције првог и другог модела лево око - x





Графикон 5.17 Предикције првог и другог модела лево око - у

На основу графикона 5.16 и 5.17, на први поглед рекло би се да није било потребе уводити другу мрежу у систем. Међутим, пажњу треба обратити да се овде ради о денормализованим вредностима оба модела и вредности су представљене у пикселима, тада разлика постаје очигледна. Други модел користи се унутар апликације за праћење пажње возача и на основу експерименталних резултата зна се да су предикције другог модела веома тачне. Разлике у предикцијама између ова два модела су у распону од 20 – 25 пиксела, што је неприхватљиво и грешка је изразито велика. Долази се до закључка да први модел мреже није способан да поуздано пронађе централне тачке левог ока, односно имало је смисла уводити други модел у систем. Слични резултати се добијају и за десно око.

### 5.2.5 Временска потрошња појединих целина програмског кода

У наредном делу, табеларно је приказана временска потрошња појединих целина програмског кода апликације за праћење пажње возача. У зависности од тога о којим деловима је реч програмски код се делом извршава на графичком процесору, а делом на централној процесорској јединици. Делови који служе за обраду и припрему података за улазе неуронских мрежа извршавају се на централној процесорској јединици, док се саме предикције извршавају на графичком процесору рачунара.

	<i>Intel i7-6700hq [ms]</i>	<i>Intel i7-8700K [ms]</i>
<b>Припрема фотографије</b>	0.3	0.1
<b>Припрема лица</b>	0.4	0.3
<b>Припрема елемената лица</b>	0.5	0.2
<b>Интерпретација</b>	5	4

Табела 5.4 Временска потрошња делова кода који се извршавају на централној процесорској јединици

	<i>Nvidia GTX960M 4GB [ms]</i>	<i>Nvidia GTX1050 2GB [ms]</i>
<b>Предикције првог модела</b>	16	10
<b>Предикције другог модела</b>	16	10
<b>Предикције трећег модела</b>	16	10

Табела 5.5 Временска потрошња делова кода који се извршавају на графичком процесору

## 6 Закључак

У овом раду описано је једно решење система за праћење пажње возача, које се као своје градивне целине, ослања на принципе дубоког учења са скупом конволутивних неуронских мрежа и приступом надгледаног учења. Неуронске мреже показале су се као ефикасно и елегантно решење полазног проблема из разлога јер ако се располаже са довољно добрим скупом података може се много тога постићи. Оне су изузетно осетљиве и са вештим руковањем могу са врло великом прецизношћу да обезбеде резултате који су потребни.

На основу приказаних резултата у поглављу Евалуација, може се закључити да ово имплементација никако не може представљати комерцијално решење, али може бити добра полазна референца. Фактори који се користе за процену присутног нивоа пажње возача нису довољни за потпуну процену, те постоји још много фактора који нису узети у обзир. Такође, проблем може представљати ноћна возња, у којој би се морало пронаћи адекватно решење извора фотографија које би омогућило неуронским мрежама несметану обраду. Важно је напоменути да су модели неуронских мрежа обучавани од почетка, односно нису претходно обучавани па само специјализовани за потребе рада. За њихово обучавање користили су се релативно мали скупови података, који нису преузети са интернета, већ генерисани имплементираним алатом. Око 10 хиљада фотографија кориштено је за обучавање првог модела и по око 5 хиљада фотографија за обучавање другог и трећег модела. Показано је да и за такве, релативно мале скупове података, неуронске мреже могу да дају довољно добре и задовољавајуће резултате.

Упркос свему томе, циљеви рада су испуњени. Омогућено је генерисање скупова података, имплементирана је апликација за означавање и генерисање *.csv* датотека жељеног и потребног формата, конструисани су модели конволутивних неуронских мрежа, обучени и примењени унутар финалне апликације. Стечено је практично искуство у раду са вештачким неуронским мрежама и одговарајућим библиотекама, манипулисањем и организацијом великих скупова података, али и другим програмским алатима и усавршена већ постојећа знања у програмским језицима *Python* и *C#*.

Добра имплементација оваквог система у аутомобилској индустрији била би од великог значаја, чији резултати би се одразили и на многе друге учеснике у саобраћају и њихову безбедност, а не само на безбедност возила у коме је имплементирано. Такође, уз минималне измене она би могла да се нађе у разним сферама, где као пример може да се узме присуство пажње студената у условима учења на даљину.

## 7 Референце

- [1] S. A. Uchil and R. Yazdanifard, "The Growth of the Automobile Industry : Toyota's Dominance in United States," *J. Res. Mark.*, vol. 3, no. 2, p. 265, 2014, doi: 10.17722/jorm.v3i2.86.
- [2] "Implications of Advances in Imaging Recognition & Image Processing Technology in Terms of Enhancing Automotive Safety."
- [3] G. L. Masala and E. Grosso, "Real time detection of driver attention: Emerging solutions based on robust iconic classifiers and dictionary of poses," *Transp. Res. Part C Emerg. Technol.*, vol. 49, no. December 2014, pp. 32–42, 2014, doi: 10.1016/j.trc.2014.10.005.
- [4] B. Alpert, "Deep Learning for Distracted Driving Detection." <https://www.nauto.com/blog/nauto-engineering-deep-learning-for-distracted-driver-monitoring>.
- [5] Y. C. A. P. Reddy, P. Viswanath, and B. E. Reddy, "Semi - supervised learning : a brief review," vol. 7, pp. 81–85, 2018.
- [6] S. S. Mousavi, M. Schukat, and E. Howley, "Deep Reinforcement Learning: An Overview," *Lect. Notes Networks Syst.*, vol. 16, pp. 426–440, 2018, doi: 10.1007/978-3-319-56991-8\_32.
- [7] M. Z. Alom *et al.*, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," 2018, [Online]. Available: <http://arxiv.org/abs/1803.01164>.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [9] BOLDENTHUSIAST, "NEURAL NETWORKS AND DEEP LEARNING: AN OVERVIEW." <https://alphabold.com/neural-networks-and-deep-learning-an-overview/>.
- [10] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017, doi: 10.1109/JPROC.2017.2761740.

- 
- [11] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.
  - [12] L. Tóth, “Phone recognition with hierarchical convolutional deep maxout networks,” *Eurasip J. Audio, Speech, Music Process.*, vol. 2015, no. 1, 2015, doi: 10.1186/s13636-015-0068-3.
  - [13] I. Wallach, M. Dzamba, and A. Heifets, “AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery,” pp. 1–11, 2015, [Online]. Available: <http://arxiv.org/abs/1510.02855>.
  - [14] M. Z. Alom *et al.*, “A state-of-the-art survey on deep learning theory and architectures,” *Electron.*, vol. 8, no. 3, pp. 1–67, 2019, doi: 10.3390/electronics8030292.
  - [15] S. Verma, “Understanding 1D and 3D Convolution Neural Network | Keras.” <https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610>.
  - [16] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.
  - [17] D. Nautiyal, “Underfitting and Overfitting in Machine Learning.” <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>.
  - [18] Ó. Cobos, J. Munilla, A. M. Barbancho, I. Barbancho, and L. J. Tardón, “Facial activity detection to monitor attention and fatigue,” *Proc. Sound Music Comput. Conf.*, pp. 295–296, 2019.
  - [19] P. Smith, M. Shah, and N. da Vitoria Lobo, “Determining driver visual attention with one camera,” *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 4, pp. 205–218, 2003, doi: 10.1109/TITS.2003.821342.
  - [20] A. Lenail, “NN-SVG.” <http://alexlenail.me/NN-SVG/index.html>.
  - [21] P. O. U. Sing and M. O. E. Nsembles, “Epo Pt : L Earning R Obust N Eural N Etwork,” vol. 1, no. 2, pp. 1–15, 2017.
  - [22] A. S. Badura-Brack *et al.*, “Effect of attention training on attention bias variability and PTSD symptoms: Randomized controlled trials in Israeli and U.S. Combat Veterans,” *Am. J. Psychiatry*, vol. 172, no. 12, pp. 1233–1241, 2015, doi: 10.1176/appi.ajp.2015.14121578.
  - [23] M. Mak *et al.*, “The efficacy of cognitive rehabilitation with RehaCom programme in schizophrenia patients. the role of selected genetic polymorphisms in successful cognitive rehabilitation,” *Ann. Agric. Environ. Med.*, vol. 20, no. 1, pp. 77–81, 2013.
  - [24] A. K. Pradhan *et al.*, “The effects of focused attention training on the duration of novice drivers’ glances inside the vehicle,” *Ergonomics*, vol. 54, no. 10, pp. 917–931, 2011, doi: 10.1080/00140139.2011.607245.
  - [25] D. Zang, Z. Chai, J. Zhang, D. Zhang, and J. Cheng, “Vehicle license plate recognition using visual attention model and deep learning,” *J. Electron. Imaging*, vol. 24, no. 3, p. 033001, 2015, doi: 10.1117/1.jei.24.3.033001.