



УНИВЕРЗИТЕТ У НИШУ
ЕЛЕКТРОНСКИ ФАКУЛТЕТ
Катедра за рачунарство



ПРАЋЕЊЕ РЕГУЛАРНОСТИ ПОЛАГАЊА ТЕСТОВА ПРИМЕНОМ ТЕХНИКА РАЧУНАРСКОГ ВИДА

- ДИПЛОМСКИ РАД -

Задатак:

Упознати се са постојећим софтверским решењима која се користе за праћење регуларности онлајн полагања тестова. Идентификовати технике рачунарског вида које се могу искористити за реализацију оваквих система. У практичном делу имплементирати прототип система који употребом камере обезбеђује препознавање особе која полаже тест, врши детекцију нерегуларности у виду окретања и разговора, те генерише извештај са видео доказом спорних ситуација.

Ментор: проф. др Александар Милосављевић **Кандидат:** Ивана Миливојевић 16704

Комисија:

1. _____	Датум пријаве: _____
2. _____	Датум предаје: _____
3. _____	Датум одбране: _____

Ниш, 2022.

Садржај

1. Увод	3
2. Софтвери за праћење регуларности полагања тестова.....	4
2.1. Аутоматско надгледање	4
3. Технике рачунарског вида за праћење регуларности полагања тестова	6
3.1. Детекција објеката.....	6
3.1.1. Viola-Jones детектор.....	7
3.1.2. HOG детектор	9
3.1.3. R-CNN детектор	12
3.1.4. YOLO детектор.....	16
3.1.5. SSD детектор.....	17
3.2. Препознавање лица.....	19
3.2.1. Eigenfaces	20
3.2.2. Fisherfaces.....	22
3.2.3. Local Binary Patterns	22
3.2.4. Дубоко учење.....	24
3.3. Одређивање положаја главе	25
3.4. Праћење погледа.....	27
3.5. Детекција говора.....	29
4. Систем за праћење регуларности полагања тестова употребом камере	30
4.1. Имплементација система	32
5. Закључак.....	42
Литература	43

1. Увод

Пандемија настала услед ширења вируса Ковид-19 утицала је на све аспекте живота људи. Када је у питању образовање, било је неопходно прилагодити начин извођења наставе како би се школска година одвијала несметано. Многе образовне институције су морале да пређу на онлајн модел одржавања наставе, те је, поред увођења новина у реализацији предвиђеног наставног програма, било неопходно наћи и адекватан начин за оцењивање ученика.

Онлајн тестирање је великој мери заступљено у пракси, пошто олакшава и убрзава процес креирања тестова, полагања, као и анализу резултата. Флексибилност и могућност рада од куће јесу разлози због којих је велики број наставника одабрао овакве тестове као меру оцено. Наравно, тиме се отвара простор за различите начине „варања“. Неке од мера које се могу применити су ограничење да сви морају полагати тест истовремено, генерисање већег броја различитих питања, приказ питања и понуђених одговора у различитом редоследу, увођење временског ограничења за одговор за свако питање, онемогућавање враћања на претходно питање итд. Међутим, ове мере некада нису довољне, јер ученици могу да пронађу начин да комуницирају са неким или потраже одговоре у литератури и на интернету. Из тог разлога направљени су софтвери за надгледање полагања онлајн тестирања. У наставку ће се особа која полаже тест оловљавати као: ученик, али ови софтвери, поред примене у образовним институцијама, имају примену и у различитим компанијама и организацијама за потребе тестирања током интервјуа за посао, реализације семинара, добијања сертификата...

У оквиру система за надгледање полагања најчешће се прате микрофон, камера и екран ученика. Једна од могућности је да дежурна особа надгледа ученике током полагања; друго решење подразумева снимање ученика током полагања и касније прегледање тих снимака ради контроле регуларности. Предност оваквог решења је што ученици и дежурна особа не морају бити присутни у заказано време, али, у овом случају у реалном времену не постоји могућност реаговања на неправилности због којих је можда требало санкционисати ученика или онемогућити му наставак полагања теста. Треће решење јесте аутоматско нагледање полагања коришћењем алгоритама вештачке интелигенције, које у великој мери мења дежурну особу и не захтева њену константну присутност. Систем приказује упозорења ученику током полагања (при чему се може алармирати надлежна особа да се прикључи полагању) и бележи све детектоване спорне ситуације које се касније могу прегледати како би се потврдиле неправилности [1].

Рад се састоји из пет поглавља. Након уводног дела, у другом поглављу је дат преглед постојећих софтверских решења за проблем праћења регуларности онлајн тестирања, са фокусом на аутоматско надгледање. У трећем поглављу дат је преглед техника рачунарског вида које се у овим софтверима могу применити за обраду података са камере. Прво су обрађене технике и алгоритми који се могу применити за детекцију објеката, на које се надовезују методе за детекцију и препознавање лица. Након тога су описани начини за одређивање положаја главе, праћење погледа и детекцију причања на видео снимку. У наредном поглављу описана је конкретна имплементација једноставног прототипа система који се може користити за аутоматско надгледање полагања тестова употребом камере. Систем врши препознавање лица и детектује нерегуларности које се најчешће могу јавити током полагања, а то су окретање и причање. Поред тога, детектује се одсуство особе у кадру или присуство више особа, коришћење мобилног телефона и гледање са стране; приказују се упозорења ученику за време теста и све детектоване спорне ситуације бележе у видео фајл. Закључно поглавље осврће се на претходна поглавља и сумира теоријске и практичне аспекте описаног проблема.

2. Софтвери за праћење регуларности полагања тестова

Постоји велики број софтвера који су развијени у циљу омогућавања реализације учења на даљину (при чему неки од њих укључују могућност полагања тестова), а доступни су и софтвери намењени само реализацији тестирања. Често је реч о апликацијама којима се приступа преко веб-претраживача, али има и апликација које је потребно инсталирати на свом уређају. Као најважнији захтеви које је потребно да ови софтвери испуне могу се издвојити: једноставност коришћења, сигурност осетљивих података (као што су сами тестови и лични подаци корисника), скалабилност, прилагодљивост и једноставна интеграција са постојећим платформама за учење. Скалабилност има важну улогу када је у питању употребљивост система, зато што може бити потребно да и више хиљада корисника истовремено полаже тестове. Систем треба да буде прилагодљив зато што корисници апликације имају различите потребе и треба им обезбедити могућност избора скупа функционалности за реализацију конкретног тестирања [2].

Када су у питању озбиљнији тестови чији интегритет треба очувати, треба одабрати (у зависности од потреба и буџета организације) неки од напреднијих софтвера за тестирање који нуде и надгледање регуларности полагања. Софтвери који врше праћење регуларности полагања тестова проширују могућности платформи за учење (и тестирање). У уводном поглављу је напоменуто да можемо разликовати три групе ових софтвера, од којих једна врши аутоматско надгледање и не захтева ангажовање дежурне особе током тестирања. Софтвери из ове групе се све више користе у пракси и омогућавају детаљнију анализу резултата, која се потом може искористити за унапређење процеса тестирања.

2.1. Аутоматско надгледање

У општем случају, у оквиру софтвера за аутоматско надгледање полагања може постојати више модула који су задужени за обраду података из различитих извора. Модул за праћење аудио података захтева да ученик има повезан микрофон и да обезбеди тишину у просторији у којој ће радити тест. Уколико тест захтева гласовну активност ученика, може се испитивати да ли је учеников глас исти током трајања теста, да ли се поклапа са гласом тог ученика из базе података, као и да ли ученик разговара са неким. Неки системи могу детектовати одређене кључне речи за које се процењује да би их ученици могли користити ако желе да потраже помоћ. Модул задужен за праћење екрана најчешће онемогућава сликање и снимање прозора у коме је отворен тест, напуштање теста и отварање других страница, апликација или фајлова. Неки од софтвера омогућавају наставницима да означе веб сајтове којима желе да омогуће приступ ученицима током полагања. Модул за обраду података са камере захтева да камера буде укључена током полагања теста и може да контролише да ли у кадру има других особа, недозвољених предмета (попут мобилних телефона, књига и папира), да врши препознавање лица, да прати покрете ученика и детектује када се он окреће или гледа са стране. Софтвер за аутоматско надгледање полагања, уз помоћ вештачке интелигенције, додатно може вршити и онлајн претрагу како би проверио да ли је садржај теста „процурео“, при чему може преузети и кораке за његово уклањање. Систем на крају полагања генерише одговарајући извештај (најчешће у видео и PDF формату) који се касније може прегледати и анализирати [3].

Honorlock, један од првих сервиса за надгледање онлајн тестирања, комбинује аутоматско надгледање са људским надгледањем – тестирање прати вештачка интелигенција, и, ако детектује нерегуларности, алармира дежурну особу да се прикључи полагању [4]. Предуслови за полагање теста су да особа буде сама у просторији, да рачунар на коме се ради тест има повезан само један монитор и да корисник поседује 360° камеру како би се скенирала просторија у којој се налази. Овај систем користи софтвер за закључавање веб претраживача који онемогућава приступ другим веб сајтовима, при чему закључава и одређене пречице на тастатури – нпр. за копирање/лепљење (енг. *copy/paste*) и сликање екрана (енг. *print screen*), онемогућава минимизирање претраживача и напуштање теста пре предаје. Такође, софтвер идентификује процедуре садржај теста на интернету и предузима кораке за његово уклањање. Још једна функционалност која је уведена јесте да систем детектује ако ученик покуша да приступи материјалу за учење током теста путем неког другог уређаја и бележи снимак екрана током трајања приступа. Овај систем не користи биометријске методе током полагања, већ пре почетка теста (коришћењем камере) услика лице ученика и његову идентификациону картицу.

Mercer / Mettl нуди још једно од решења које омогућава безбедно тестирање, и може се користити као комплетна платформа за тестирање, или као сервис за удаљено надгледање тестирања којим се проширује нека платформа за тестирање [5]. Систем детектује уколико кандидат није присутан или је присутна особа која није она којом се представља (током трајања теста се у одређеним временским интервалима врши препознавање лица), присуство мобилних телефона и разговор са другим особама. Тестови се могу полагати и на уређајима попут мобилних телефона и таблета, али тестове који захтевају већу сигурност неопходно је полагати на лаптоп или десктоп рачунарима. Још једна од функционалности овог система је могућност праћења локације са које се полаже тест. У понуди је и безбедни претраживач који искључује све екстерне портове, апликације за дељење података и веб сајтове. Систем је обучен да детектује до 18 типова нерегуларности.

3. Технике рачунарског вида за праћење регуларности полагања тестова

Рачунарски вид је област рачунарске науке у којој се разматрају могућности да се уз помоћ рачунара уоче (виде), идентификују и обраде дигиталне слике на сличан начин како то чини људски вид. Притом се тежи разумевању и аутоматизацији задатака које може да обради човеков визуелни систем. Као научна дисциплина, рачунарски вид се бави теоријом вештачких система која издваја информације из слика, док као технолошка дисциплина настоји да примени теорије и моделе за креирање система рачунарског вида [6].

Када је реч о апликацијама за праћење полагања онлајн тестова, технике рачунарског вида можемо применити у оквиру модула који обрађује податке са камере. Неке од техника које се могу применити су: детекција објеката, детекција и препознавање лица, одређивање положаја главе, праћење погледа и детекција причања особе која полаже тест.

3.1. Детекција објеката

Детекција објеката је рачунарска технологија која спада у област рачунарског вида и бави се проналажењем објеката који припадају одређеним класама на дигиталним сликама и видео-снимцима. Свака класа објеката има специфичне особине – фичере (енг. features), који одређују припадност објекта тој класи. Поред информације о класи објекта, као резултат се добија и информација о његовој локацији на слици [7]. Детекција објеката има широку примену у свакодневном животу (нпр. у саобраћају: за проналажење пешака, аутомобила и саобраћајних знакова), а један од најпознатијих специфичних случајева детекције објеката јесте детекција лица.

Методе за детекцију објеката се могу поделити на оне које се заснивају на традиционалним техникама обраде слика и оне које су базиране на дубоком учењу. Код метода које су засноване на традиционалним техникама издвајају се фичери објеката коришћењем неког од детектора фичера и након тога се користе технике за класификацију. Технике базиране на дубоком учењу се ослањају на конволуционе неуронске мреже. Поступак детекције код њих се може реализовати кроз две фазе или у оквиру једне фазе. Код двофазних детектора прва фаза јесте проналажење произвољног броја потенцијалних региона у којима се налазе објекти, а друга је класификација сваког од њих, уз одређивање коначног правоугаоника којим се објекат може уоквирити. Једнофазни детектори предвиђају оквирне правоугаонике објеката без претходно издвојених региона од интереса третирајући детекцију објеката као регресиони проблем. Регресија оквирних правоугаоника (енг. bounding box regression) је кључни концепт у предвиђању њихових координата. Ови детектори су бржи и структурно једноставнији, али су и нешто мање поуздани од двофазних. Раније методе за детекцију могу бити ограничене услед постојања комплексне позадине, делимично прекривених објеката и лошег осветљења, док су технике дубоког учења значајно отпорније на наведене проблеме. Дубоке конволуционе неуронске мреже у комбинацији са убрзањем које доноси графички процесор дају веома добре резултате и омогућавају детекцију објеката у приближно реалном времену. Недостатком ових метода може се сматрати потреба за значајним хардверским ресурсима и великим бројем тренинг слика које треба ручно означити [8].

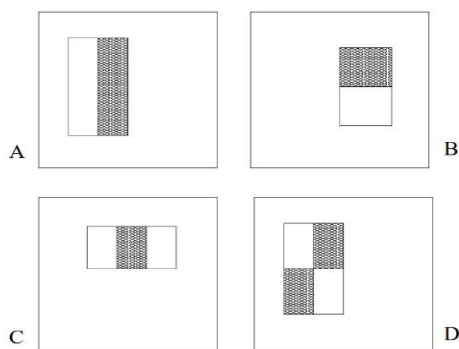
Када је у питању евалуација детектора објеката, најпопуларнији референтни скупови података су COCO и PASCAL VOC. Метрике које се најчешће користе су Frames Per Second (FPS) и mean Average Precision (mAP). FPS представља број обрађених фрејмова у секунди, а mAP средњу просечну прецизност чије се рачунање мало разликује за COCO и PASCAL VOC, тако да при упоређивању ове мере треба узети у обзир над којим скупом је рачуната. У општем случају, просечна прецизност (AP) се рачуна за сваку класу објеката на основу графика зависности прецизности (енг. precision) од комплетности (енг. recall). Прецизност представља проценат позитивно класификованих података који су заиста позитивни, док комплетност представља проценат исправно класификованих позитивних података у односу на укупан број позитивних података. После тога се налази mAP као средња вредности израчунатих AP [9]. Тежи се што већој mAP мери и што већем FPS.

У наставку ће бити дат преглед неколико детектора објеката који су били важни за развој ове области, почевши од Viola-Jones и HOG детектора – који се заснивају на традиционалним техникама, до двофазних (различите варијанте R-CNN-a) и једнофазних (YOLO и SSD) детектора – који користе конволуционе неуронске мреже.

3.1.1. Viola-Jones детектор

Paul Viola и Michael Jones (у раду *Rapid Object Detection using a Boosted Cascade of Simple Features* [10] из 2001. године) представљају метод за детекцију објеката у реалном времену реализован кроз примену неколико нових техника. Овај метод је првобитно био намењен детекцији лица, али се може користити и за детекцију било ког типа објеката. Viola-Jones детектор се често назива и „Haar cascade“, зато што се за детекцију објеката користе каскада класификатора и скуп фичера који подсећају на Haar-ове таласе.

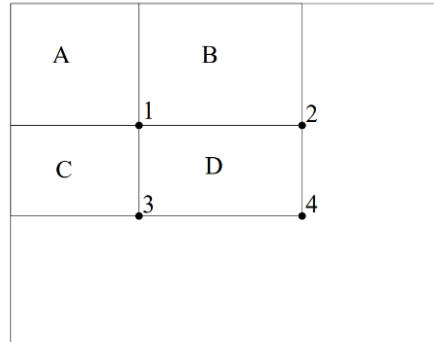
Аутори користе скуп фичера приказан на слици 1. Притом, дефинисане су три врсте фичера: фичери са два (A и B), три (C) и четири (D) суседна правоугаоника. Вредност фичера се рачуна тако што се сума пиксела у белим регионима одузима од суме пиксела у сивим. Ови фичери омогућавају проналажење ивица и линија на слици.



Слика 1: Haar фичери [10]

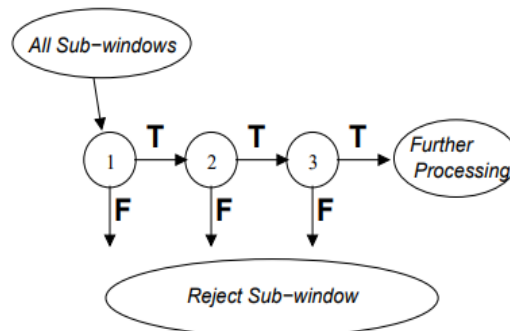
У раду је приказан нов начин за репрезентацију слике, назван „интегрална слика“, који омогућава брзо рачунање фичера. Интегрална слика се рачуна једним проласком кроз оригиналну слику и може се посматрати као матрица истих димензија као оригинална слика, с тим што на локацији (x, y) садржи суму пиксела оригиналне слике изнад и лево од тачке (x, y). Оваква репрезентација омогућава израчунавање суме пиксела произвољног правоугаоног региона коришћењем вредности у само 4 тачке, односно, применом само 3 операције сабирања/одузимања. На слици 2 се може приметити да сума пиксела у оквиру правоугаоника D може бити израчуната на основу

вредности интегралне слике у тачкама 1, 2, 3 и 4. Вредност интегралне слике у тачки 1 је сума пиксела у правоугаонику A. Вредност у тачки 2 је $A + B$, у тачки 3 је $A + C$, а у тачки 4 је $A + B + C + D$. Сума пиксела у оквиру правоугаоника D би била: $4 + 1 - (2 + 3)$. За два правоугаоника сума пиксела може бити израчуната на основу вредности у 6 тачака, за 3 правоугаоника на основу 8, а за случај 4 правоугаоника на основу 9 тачака. Када се интегрална слика једном израчуна, фичери било које величине на било којој локацији могу се израчунати за константно време.



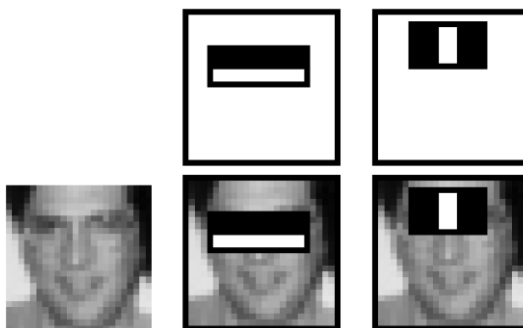
Слика 2: Рачунање суме пиксела у оквиру правоугаоног региона [10]

За прозор димензија 24×24 се израчунавају фичери (око 180 000 фичера различитих величина и позиција). Иако се појединачни фичери израчунавају прилично једноставно и брзо, рачунање комплетног скупа фичера је захтевно и скупо. Да би издвојили мањи скуп најважнијих фичера из великог скупа фичера и тренирали класификатор, **Viola** и **Jones** су користили алгоритам заснован на AdaBoost (Adaptive Boosting) техници. Свака фаза *boosting* процеса даје нови слаби класификатор, који зависи само од једног фичера (бира се фичер који има најмању грешку класификације). Овиме се број фичера смањује на око 6000. AdaBoost алгоритам комбинује више слабих класификатора (који не могу самостално да класификују слику) у јак (добар) класификатор. За детекцију објеката је коришћена каскада јаких класификатора који се примењују један за другим, при чему је сваки класификатор у каскади комплекснији од претходног. На слици 3 је илустрован процес класификације. Ако прозор не прође први класификатор (не садржи објекат), одбацује се и не процесира се даље. Ако прође, пропушта се кроз наредни класификатор и процес се понавља. Прозор који прође све класификаторе сматра се прозором који садржи објекат. Ова каскада личи на дегенерисано стабло одлучивања код кога за сваки „родитељски чвор“ постоји само једно „дете чвор“. Циљ је брзо одбацити регионе без објеката и даље процесирање вршити само над регионима који су обећавајући, односно онима који потенцијално садрже објекат.



Слика 3: Каскада класификатора [10]

Детекција објеката је демонстрирана на примеру детекције лица. На слици 4 приказана су прва два фичера селектована AdaBoost алгоритмом. Први фичер наглашава особину лица да је регион око очију углавном тамнији од региона горњег дела образа, док други фичер осликава особину да је регион очију тамнији од региона носа. Сваки од класификатора у каскади од 38 слојева трениран је фронталним сликама лица које су скалиране на резолуцију 24×24 (како би се поклопиле са прозором) и сликама на којима нема лица, са којих су издвајани подпрозори 24×24 . Број фичера у првих пет слојева је 1, 10, 25, 25 и 50, респективно. У процесу детекције детектор обилази слику која се тестира у више величина методом клизајућег прозора, при чему се детектор (прозор) скалира, а не слика, како би се искористиле предности интегралне слике. С обзиром на то да ће бити генерисано више детекција за исто лице, потребно је на крају те детекције свести на једну. Све детекције се раздвајају у непреклапајуће подскупе, при чему ће се две детекције наћи у истом подскупу ако им се оквирни региони преклапају. Сваки подскуп даће једну коначну детекцију чије границе су просек граница свих детекција у том подскупу.



Слика 4: Прва два фичера селектована AdaBoost техником [10]

Viola-Jones детектор ради са сивим (енг. grayscale) сликама, детектује објекте на сликама независно од њихове локације и величине и инваријантан је на промене у осветљењу. Иако је објављен пре много година (2001), он се и даље често користи за детекцију лица. Једна од највећих предности овог детектора у односу на остале јесте велика брзина детекције (од 15 FPS, у периоду када је објављен), а недостатак представља склоност лажним позитивним детекцијама и детектовање једино фронтално постављених објеката.

3.1.2. HOG детектор

У раду *Histograms of Oriented Gradients for Human Detection*, објављеном 2005. године [11], Dalal и Triggs су показали да се за детекцију објеката на слици могу успешно користити хистограм оријентисаних градијената (енг. Histogram of Oriented Gradients, HOG) и линеарни SVM класификатор. Аутори су се фокусирали на детекцију пешака на слици, али се детектор може применити и за детекцију објеката других класа.

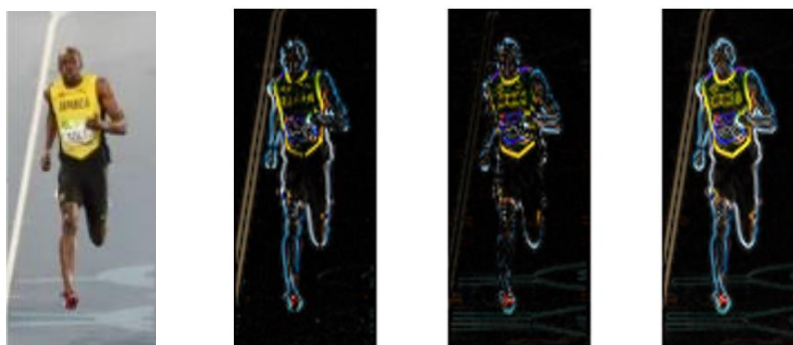
Фичер дескриптор је репрезентација слике којом се издвајају корисне информације из ње, а одбацују ирелевантне. Код HOG фичер дескриптора се улазна слика димензија $64 \times 128 \times 3$ (канала) конвертује у вектор фичера дужине 3780, при чему се као фичери користе хистограми смерова оријентисаних градијената. Градијенти слике (x и y изводи) су корисни зато што је њихова магнитуда велика око ивица и ћошкова (региони наглих промена интензитета), а они носе много више информација о облику објекта него равни региони. Први корак у алгоритму је препроцесирање слике. У оригиналном раду HOG дескриптор фичера је рачунат над регионима слике величине

64×128, али у општем случају слика може бити било које величине. Једино ограничење је да регион који се обрађује има фиксни однос ширине и висине (у овом случају 1:2). Након тога је потребно израчунати хоризонталне и вертикалне градијенте, што се може постићи филтрирањем слике Собел филтером. Магнитуда и угао градијената се рачунају по следећим формулама:

$$G = \sqrt{G_x^2 + G_y^2}, \quad (1)$$

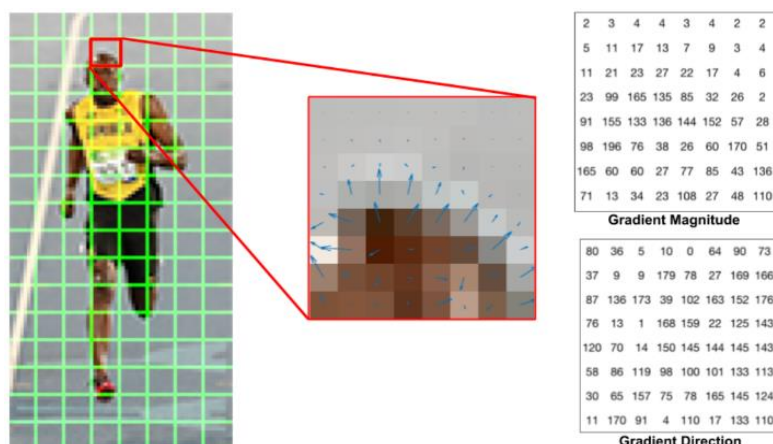
$$\theta = \tan^{-1} \frac{G_y}{G_x}, \quad (2)$$

где су G_x и G_y градијенти по x и y осама. На слици 5 у левом делу приказана је слика особе која трчи, у средишњем делу су приказани x градијент (који издваја вертикалне линије) и y градијент (који издваја хоризонталне линије), док је финална слика приказана десно. Особа је у првом плану, а већина неважних информација је уклоњена. Код слика у боји рачунају се градијенти за сва три канала, па се за магнитуду у сваком пикселу узима максимална магнитуда та три градијента, а за угао онај угао који одговара максималној магнитуди.



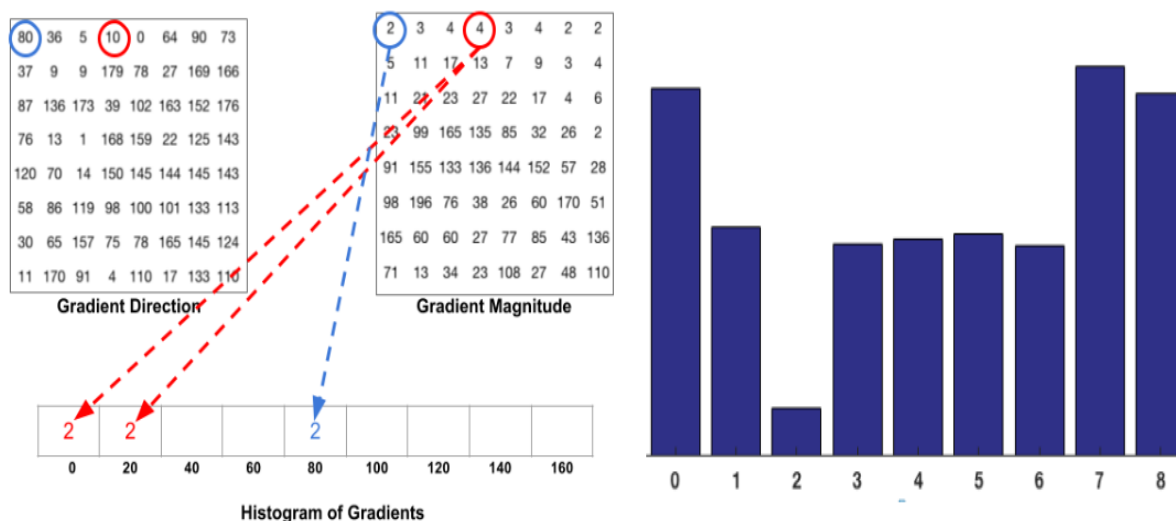
Слика 5: Део улазне слике (лево), његови x и y градијенти (средина) и финална магнитуда градијената (десно) [12]

Затим се слика дели на ћелије од 8×8 пиксела и за сваку ћелију се рачуна хистограм градијената. Ћелија садржи $8 * 8 * 3 = 192$ вредности пиксела, а градијент овог дела слике садржи 2 вредности (магнитуду и смер) за сваки пиксел, што даје $8 * 8 * 2 = 128$ вредности. Индивидуални градијенти пиксела могу садржати шум, али хистограм над ћелијом од 8×8 пиксела је много отпорнији на шум.



Слика 6: Подела слике на ћелије (лево), градијенти ћелије представљеним стрелицама (средина), градијенти ћелије представљени бројевима (десно) [12]

На слици 6 у левом делу приказана је подела слике на ћелије. У средини је издвојена једна ћелија чији су смерови градијената илустровани смером стрелица, а магнитуде дужинама стрелица. Смерови стрелица показују смер промене интензитета, а њихова дужина колико је велика разлика. Десно су бројевима представљене вредности магнитуда и смерова градијената за пикселе издвојене ћелије. Углови се посматрају у односу на у осу и у опсегу су између 0° и 180° уместо 0° и 360° зато што су посматрани „неозначени“ градијенти.

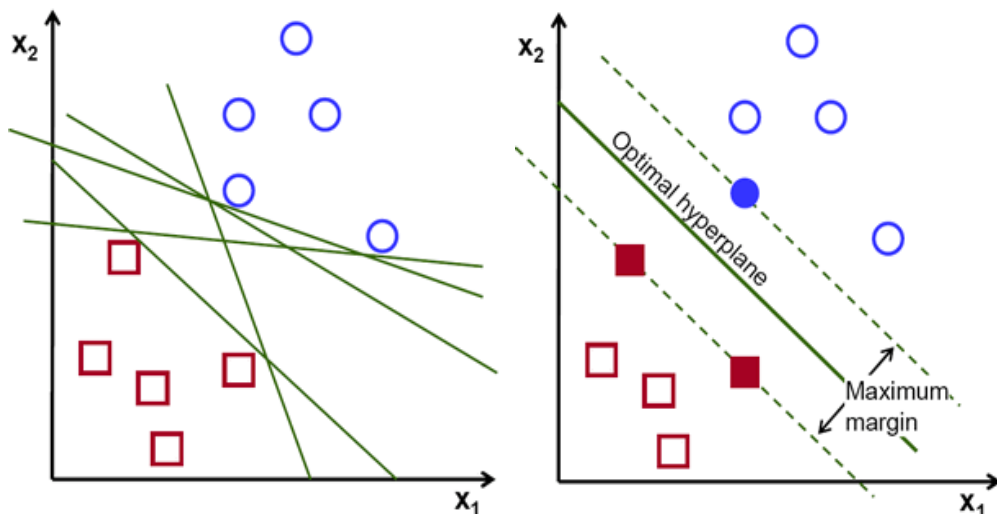


Слика 7: Процес рачунања хистограма градијената (лево),
график хистограма (десно) [12]

Следећи корак је рачунање хистограма у оквиру ових ћелија (илустровано на слици 7 у левом делу). Хистограм је низ од 9 елемената који одговарају угловима од 0° , 20° , 40° , ..., 160° . На основу смера (угла) градијента бира се елемент низа, а магнитуда утиче на вредност одабраног елемента. Градијент означен плавом бојом има угао од 80° и магнитуду 2, тако да се вредност 2 додаје елементу на 5. позицији. Градијент означен црвеном бојом има угао од 10° који је на средини између 0° и 20° , тако да се његова вредност 4 додаје пропорционално 1. и 2. елементу (по 2). Ако је угао већи од 160° , он је заправо између 160° и 180° , а како су углови од 0° и 180° еквивалентни, његова вредност би била пропорционално придружена елементима придруженим угловима од 0° и 160° . У десном делу слике 7 је приказано како би изгледао хистограм за одабрану ћелију са слике 6. Како у хистограму доминирају вредности близу 0° тј. 180° , може се закључити да ова ћелија вероватно садржи ивицу јер су градијенти усмерени на горе или доле. Да би дескриптор био инваријантан на промене у осветљењу, треба нормализовати¹ хистограм. Посматра се прозор 16×16 који садржи 4 хистограма и нормализација се врши над вектором од 36 елемената који је формиран надовезивањем ових хистограма. Прозор се затим помера за 8 пиксела и процес се понавља док се не обиђе цела слика. Да би се израчунао коначни вектор фичера за комплетно парче слике, нормализовани вектори дужине 36 се конкатенирају и креирају један велики вектор од 3780 елемената ($36 * 105 = 3780$, где је $7 * 15 = 105$ број позиција на којима може да се нађе прозор). Добијени вектор се користи за тренирање SVM класификатора [12].

¹ Нормализација вектора у општем случају подразумева дељење сваке вредности вектора са дужином вектора.

Машина са векторима подршке (енг. Support Vector Machine, SVM) је техника класификације чији је циљ пронаћи оптималну хиперраван (границу одлуке) у вишедимензионалном простору, која раздваја податке који припадају различитим класама. Маргина представља најкраће растојање од хиперравни до најближег тренинг податка који припада било којој класи. Уколико није могуће у потпуности поделити податке, треба наћи хиперраван која максимизује маргину, односно минимизује грешке у класификацији. На слици 8 је у левом делу приказано више могућих хиперравни у 2D простору, а у десном оптимална хиперраван. Очекује се да хиперраван са већом маргином буде прецизнија приликом класификације непознатих података у односу на хиперраван са мањом маргином. Димензионалност простора је одређена бројем фичера [13]. Подаци који се налазе на самим маргинама називају се „вектори подршке“ (енг. support vectors) и они се најтеже класификују, али дају највише података о самој класификацији. У случају скупа података који се не може линеарно раздвојити, може се извршити трансформација тог скупа података у вишедимензионални простор где подаци јесу линеарно сепарабилни. Како се не би експлицитно примењивала функција трансформације над сваком податком, користе се кернел функције („кернел трик“), које обезбеђују нелинеарну границу у оригиналном простору.



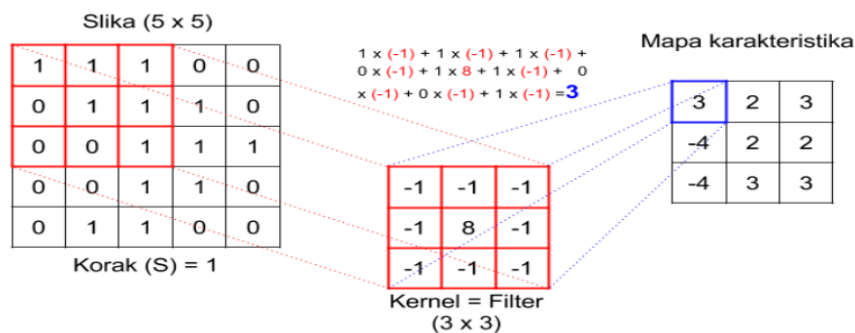
Слика 8: Могуће хиперравни (лево), оптимална хиперраван (десно) [13]

HOG фичер дескриптор у комбинацији са SVM класификатором може се користити за детекцију објеката у реалном времену, па налази примену и данас: даје добре резултате и инваријантан је на промене у осветљењу, али није инваријантан на промене у ротацији и углу гледања, због чега не може детектовати објекте који нису постављени приближно фронтално. Како би се детектовали објекти различитих величина примењују се пирамида слика и метода клизајућег прозора.

3.1.3. R-CNN детектор

R-CNN (енг. Region-based Convolutional Neural Networks) детектор, објављен 2014. године ([Girshick](#) и сар. у раду *Rich feature hierarchies for accurate object detection and semantic segmentation* [14]), један је од првих модела којим се за детекцију објеката примењује дубоко учење, а може се користити и за семантичку сегментацију. Заснива се на конволуционим неуронским мрежама, које се по својој архитектури могу сврстати у дубоке неуронске мреже. Ове мреже имају велику примену у домену рачунарског вида, зато што су пројектоване за рад са 2D структурама (као што су слике).

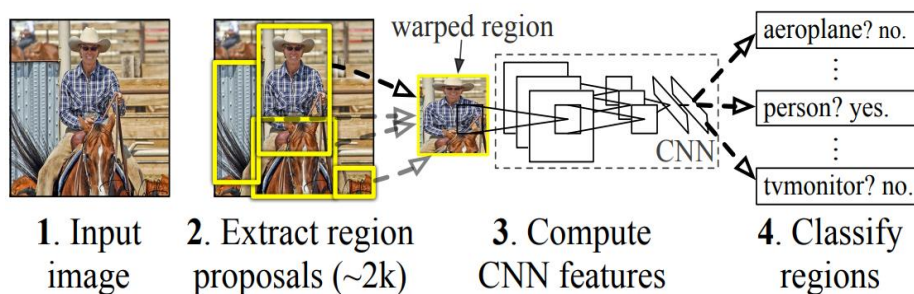
Конволуционе неуронске мреже (енг. Convolutional Neural Networks, CNN) су вештачке неуронске мреже које су добиле име по конволуцији, оператору који се примењује у обради слика за изоштравање и замућење, као и за детектовање ивица. Конволуциони филтери (кernels) се примењују на слике како би се извукли фичери и креирале њихове мапе, при чему се може вршити и редукција резолуције слика. Филтер се представља дводимензионалном матрицом малих димензија (најчешће 3×3) која се примењује у конволуционом слоју мреже на слику која се обрађује у том слоју. На слици 9 може се видети примена 3×3 филтера са кораком померања 1 на монохроматску слику димензија 5×5. У сваком кораку се вредности на истој позицији множе и сабирају са производима других парова у прозору, чиме се добија елемент мапе фичера. Архитектура мреже је таква да се на почетку налази улазни слој којим се слика уводи у мрежу. Затим следи један или више конволуционих слојева између којих се могу наћи слојеви сажимања (енг. pooling layer) који се користе са циљем прогресивног смањења слике. Такође, постоје слојеви активационе функције (најчешће ReLU), који могу значајно побољшати перформансе мреже и који се често убацују након сваког конволуционог слоја. На крају, опционо, може постојати један или више потпуно повезаних (енг. fully connected) слојева. Неурони из потпуно повезаног слоја су повезани са свим неуронима из претходног слоја [15].



Слика 9: Пример примене конволуционог филтера [15]

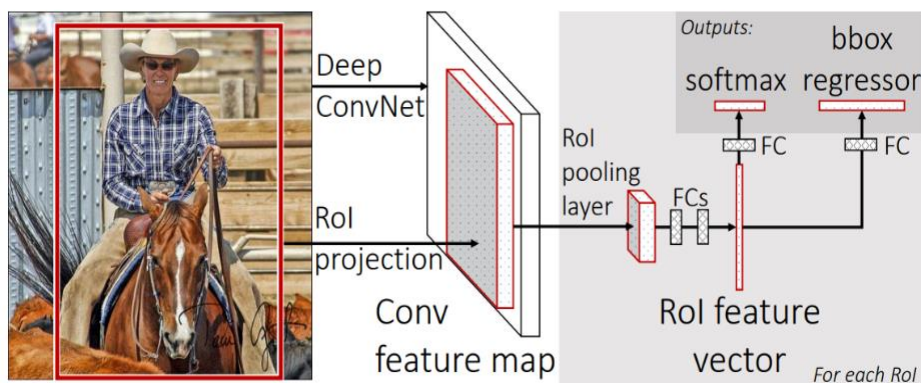
На слици 10 је приказано како R-CNN модел функционише. Girshick и сар. су направили систем за детекцију објеката који се састоји из три модула. Први модул применом селективног тражења издваја из улазне слике око 2000 региона који представљају кандидате за детекцију објеката. Издвојени региони су правоугаоници за које постоји вероватноћа да садрже објекат (без информације о томе којој класи припада тај објекат). Алгоритам селективног тражења врши сегментацију слике на основу интензитета пиксела и генерише регионе груписањем мањих сегмената сличних по боји, текстури, величини и облику. Други модул је конволуциона неуронска мрежа (позната као „AlexNet“) са 5 конволуционих слојева и 2 потпуно повезана слоја која издваја фичере (векторе фичера фиксне дужине од 4096 елемената) из сваког предложеног региона. Величина региона је пре пропуштања кроз мрежу промењена тако да одговара улазу који мрежа очекује. Трећи модул је скуп SVM-а за сваку појединачну класу објеката уз помоћ којих се класификују региони. Да би се побољшале перформансе локализације коришћена је регресија оквирних правоугаоника. Након тога, примењује се пост-процесирање да се пречисте оквирни правоугаоници и елиминише вишак. Ово се постиже коришћењем технике потискивања не-максимума (енг. non-maximum suppression), за сваку класу посебно, која одбацује правоугаоник ако има IOU² (Intersection Over Union) већи од одређеног прага са правоугаоником који има већу поузданост детекције [14].

² IOU представља однос између пресека и уније два правоугаоника.



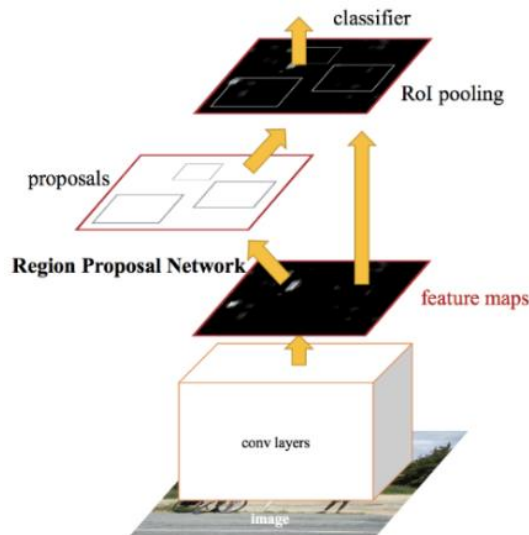
Слика 10: Начин функционисања R-CNN модела [14]

Процеси тренирања и тестирања R-CNN модела трају дуго, због чега је он веома спор. **Girshick**, један од аутора R-CNN-a, објављује 2015. године његово побољшање – Fast R-CNN [16]. Коришћена је притом VGG-16 неуронска мрежа, која има тринаест конволуционих слојева и три потпуно повезана слоја. На слици 11 приказан је начин рада Fast R-CNN-a. Мрежа која као улаз има комплетну слику и регионе генерисане селективним тражењем на основу слике генерише конволуциону мапу фичера (генерише се једна мапа фичера, а не 2000). Затим, за сваки регион, RoI (Region of Interest) слој сажимања примењује сажимање максимумом и „пакује“ регион у малу мапу фичера фиксних димензија, коју потпуно повезани слојеви (FCs) мапирају у вектор фичера фиксне дужине. Потпуно повезани слојеви се гранају у два излазна слоја: један који генерише *softmax* вероватноћу над K класа објеката (+1 за позадинску класу) и други, који генерише одступања од локација и димензија предложених региона за сваку од класа. *Softmax* функција рачуна расподелу вероватноћа припадања објекта класама, тако да свака вероватноћа буде из опсега [0, 1], а сума свих вероватноћа буде 1. Овај модел има mAP од 66.9%, док R-CNN има 66.0% на PASCAL VOC 2007.



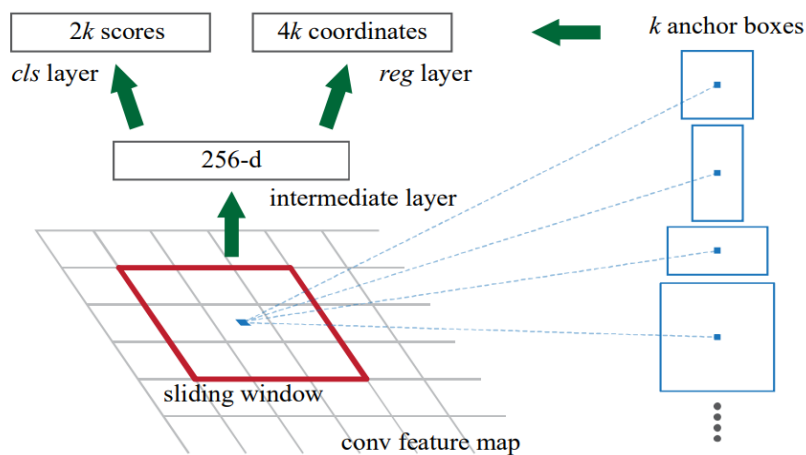
Слика 11: Архитектура Fast R-CNN-a [16]

Претходна два детектора користила су селективну претрагу за проналажење региона која је веома спора, односно представља уско грло у систему. Уместо тога, **Ren** и **cap.** су представили Faster R-CNN, у раду *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* [17], који користи дубоку конволуциону мрежу за предлагање региона. На слици 12 се може видети како изгледа архитектура комплетног Faster R-CNN система. Један модул је RPN (Region Proposal Network), мрежа која на основу конволуционе мапе фичера издваја предложене регионе. RPN је потпуно повезана конволуциона мрежа која симултано предвиђа границе објеката и информацију о вероватноћи да тај део слике садржи објекат. Други модул је Fast R-CNN детектор који користи предложене регионе. Ова два модула деле конволуционе слојеве – систем је јединствена неуронска мрежа за детекцију објеката.



Слика 12: Faster CNN детектор [17]

Принцип рада RPN је приказан на слици 13. Мрежа као улаз има $n \times n$ клизећи прозор над мапом фичера (у оригиналном раду је узето $n = 3$). Сваки прозор се мапира у 256-димензиони фичер и затим прослеђује слоју за регресију (reg) и слоју за класификацију (cls). На свакој локацији клизећег прозора предвиђа се максимално k региона, тако да регресиони слој који даје координате региона има $4k$ излаза, а класификациони који даје вероватноћу за присуство или одсуство објекта у региону има $2k$ излаза. Аутори ове регионе посматрају релативно у односу на k референтних правоугаоника названих „anchors“ који су центрирани у прозору и имају придружен фактор скалирања и однос ширина/висина (енг. aspect ratio). Подразумевано се генерише 9 оваквих правоугаоника (користећи 3 фактора скалирања и 3 aspect ratio-a). Мрежа је имплементирана кроз $n \times n$ конволуциони слој који прате два 1×1 конволуциона слоја. Овај приступ је инваријантан на транслацију тако да ће бити генерисан исти предлог и када се објекат помери на другу локацију.

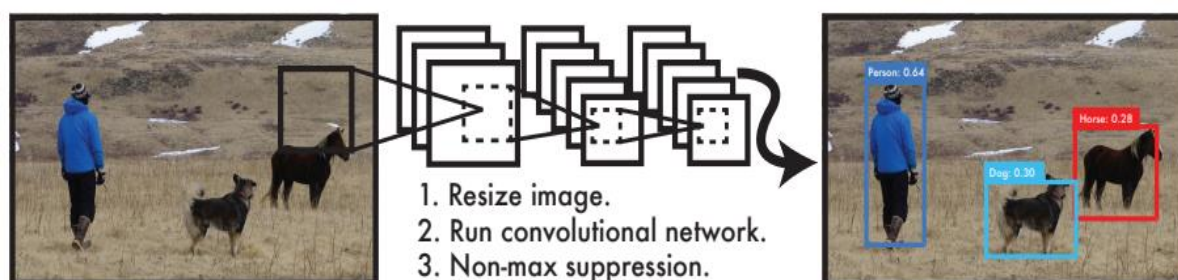


Слика 13: Region Proposal Network (RPN) [17]

Mask R-CNN [18] детектор из 2017. године је унапређење Faster R-CNN-a. Разлика између њих је у томе што Mask R-CNN додаје грану за предикцију маске објекта. Маска која се рачуна на нивоу пиксела омогућава нам да раздвојимо објекат од позадине примењујући сегментацију инстанци (енг. instance segmentation).

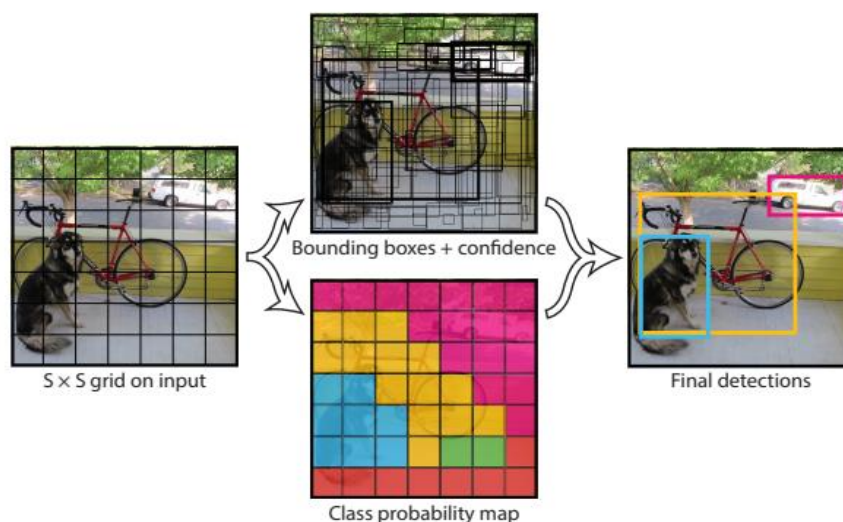
3.1.4. YOLO детектор

YOLO (енг. You Only Look Once) детектор презентован у раду Redmon-а и сар. *You Only Look Once: Unified, Real-Time Object Detection*, објављеном 2016. године [19], представљао је нови приступ проблему детекција објеката. Неуронска мрежа, инспирисана GoogleNet моделом, предвиђа оквирне правоугаонике и одговарајуће вероватноће директно из комплетних слика – отуд и потиче назив детектора: You Only Look Once („погледај само једном“).



Слика 14: Процес детекције YOLO детектором [19]

На слици 14 приказан је процес детекције YOLO детектором. Први корак је промена величине улазне слике на 448×448 пиксела, затим се таква слика пропушта кроз конволуциону неуронску мрежу, након чега се врши потискивање не-максимума, односно, одбацавање оквирних правоугаоника који су вишак. Прво се елиминишу они правоугаоници који имају поузданост мању од задате границе (0.5), а након тога се бира правоугаоник са највећом поузданошћу. Правоугаоници који имају IOU (са њим) већи од граничног биће одбачени. Процес се понавља све док сви правоугаоници не буду обрађени.



Слика 15: Модел YOLO детектора [19]

Илустрација модела овог детектора приказана је на слици 15. Аутори су делили улазну слику на грид димензија $S \times S$. Ако центар објекта упада у ћелију грида, та ћелија ће бити задужена за његову детекцију. Свака ћелија предвиђа B обухватајућих правоугаоника, при чему сваки обухватајући правоугаоник има 5 предикција: x , y , w , h , и confidence. Координате (x , y) представљају центар правоугаоника релативно у односу

на границе ћелије. Ширина и висина (w и h) представљене су релативно у односу на целу слику. Поузданост (confidence) се рачуна као $P(\text{Object}) * \text{IOU}$, где је $P(\text{Object})$ вероватноћа да правоугаоник садржи објекат, а IOU осликава тачност граница правоугаоника (рачуна се за предиктовани правоугаоник и ручно означени правоугаоник који означава где се објекат налази). Свака ћелија грида такође предвиђа C условних вероватноћа $P(\text{Class}_i | \text{Object})$. Ово су вероватноће појављивања објеката сваке од класа у GRID ћелији. Предвиђа се само један скуп ових вероватноћа по ћелији, независно од броја обухватајућих правоугаоника B . Условне вероватноће припадања класи и индивидуалне вероватноће за сваки обухватајући правоугаоник се множе, што нам даје вероватноћу припадања класи за сваки обухватајући правоугаоник:

$$P(\text{Class}_i | \text{Object}) * P(\text{Object}) * \text{IOU} = P(\text{Class}_i) * \text{IOU}. \quad (3)$$

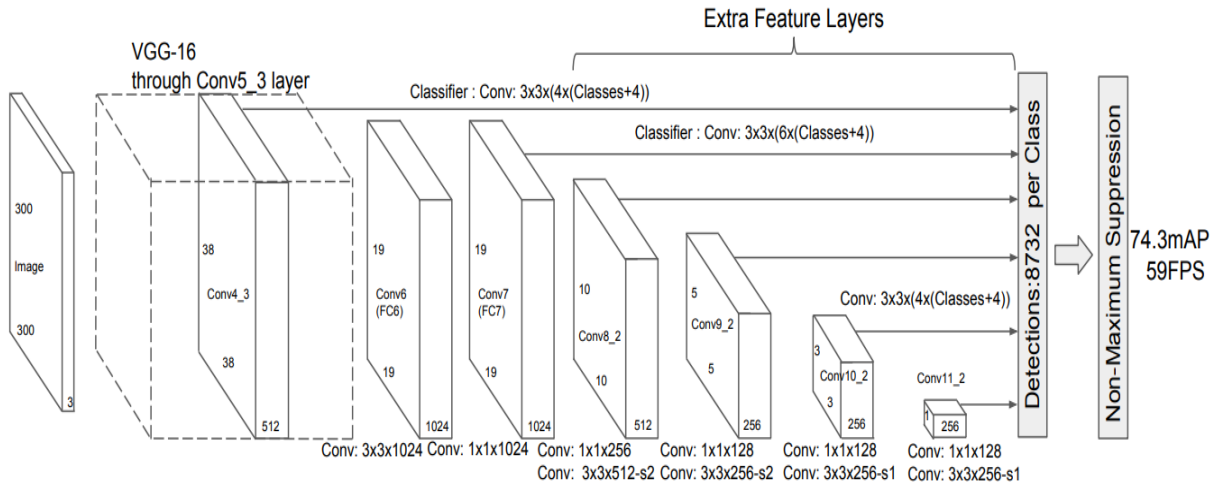
Ово нам даје вероватноћу да се класа i појављује у правоугаонику, као и информацију о томе колико добро правоугаоник представља објекат. Предикције се кодирају као $S \times S \times (B * 5 + C)$ тензор. У оригиналном раду је узето $S = 7$, $B = 2$ и $C = 20$. Модел је имплементиран као конволуциона неуронска мрежа која има 24 конволуциона слоја и 2 потпуно повезана слоја, док бржа варијанта мреже има мање конволуционих слојева (9) и мање филтера у тим слојевима. Иницијални конволуциони слојеви издвајају фичере из слике, док потпуно повезани слојеви предвиђају излазне вероватноће и координате. Модел обрађује 45 фрејмова у секунди достижући mAP од 63.4% на PASCAL VOC 2007, док Faster R-CNN достиже mAP од 73.2%, али уз 7 FPS. Бржа варијанта YOLO модела (Fast YOLO) обрађује и до 155 фрејмова у секунди (уз mAP од 52.7%) [19].

Архитектура овог детектора је била знатно бржа од оних које су нудили остали детектори који раде у реалном времену, као што је Faster R-CNN. Још неке од његових предности су што генерише мало лажно позитивних предикција када нема ничега на слици и што даје добре резултате при детекцији објеката и на уметничким сликама. Главни недостаци овог алгоритма су што не детектује добро мале објекте и објекте који су блиско груписани (зато што дели слику на GRID у коме је свака ћелија задужена за детекцију само једног објекта). У наредним годинама су објављиване новије верзије овог детектора од којих је последња YOLOv7, објављена ове године. YOLOv7 се сматра најбољим детектором у овом тренутку.

3.1.5. SSD детектор

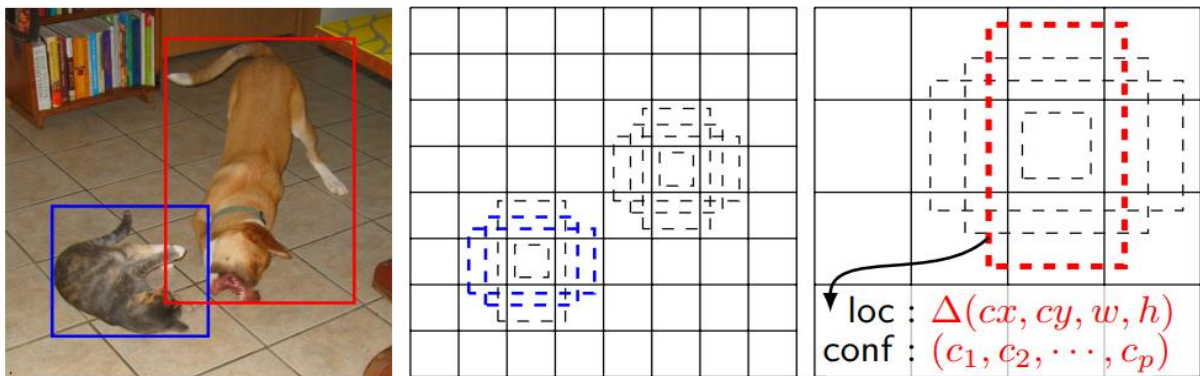
SSD (енг. Single Shot MultiBox Detector) детектор представили су Liu и сар. у раду *SSD: Single Shot MultiBox Detector* [20] 2016. године. Детекција објеката обавља се коришћењем дубоке конволуционе неуронске мреже једним проласком кроз мрежу, а дизајниран је тако да ради у реалном времену. Овај детектор уводи неке новине, као што су мапе фичера различитих величина и подразумевани правоугаоници (енг. default boxes).

Коришћена архитектура детектора приказана на слици 16 састоји се из два дела и заснована је на VGG-16 мрежи (уместо ње могу се користити и друге мреже као основа). Процес детекције обухвата издвајање мапа фичера и примену конволуционих филтера како би се детектовали објекти. VGG-16 мрежа је одсечена пре класификационог слоја и њени потпуно повезани слојеви 6 и 7 су конвертовани у конволуционе слојеве (Conv6 и Conv7). Затим следи скуп додатних конволуционих слојева (Conv8_2, Conv9_2, Conv10_2 и Conv11_2), при чему се резолуција слојева прогресивно смањује. Почетни конволуциони слојеви доприносе детекцији мањих објеката, а крајњи детекцији већих објеката.



Слика 16: Архитектура SSD детектора [20]

У конволуционом слоју Conv4_3 слика се дели на 38×38 ћелија, при чему се за сваку од њих генеришу 4 подразумевана правоугаоника, а потом се за сваки од њих генерише по једна предикција, тако да имамо укупно $38 * 38 * 4$ предикција. Генерисање више предикција се назива multibox. Предикција се састоји од предиктованог оквирног правоугаоника и 21 вредности, које означавају припадност свакој од класа (20 класа за објекте и једна класа за позадину). Предиктован оквирни правоугаоник је дефинисан релативно у односу на подразумевани правоугаоник коришћењем 4 вредности: (Δx , Δy , Δw , Δh) које представљају одступања у односу на центар (c_x , c_y), ширину (w) и висину (h) подразумеваног правоугаоника. За класу објекта узима се класа са највећом вредношћу. Рачунање предикција се обавља применом 3×3 конволуционог филтера над сваком ћелијом. Сличан поступак се обавља у сваком конволуционом слоју. Са смањењем величине слојева смањује се и резолуција мапа фичера, што омогућава детекцију објеката различитих величина и значајно побољшава тачност детектора. Укупно има 6 конволуционих слојева, при чему се у неким од њих генерише 4, а у неким 6 предикција.



Слика 17: Слика са GT правоугаоникима (лево), 8×8 мапа фичера (средина), 4×4 мапа фичера (десно) [20]

SSD дефинише фактор скалирања за сваки од слојева. Комбиновањем ових фактора скалирања и жељеног *aspect ratio*-а израчунавају се ширина и висина подразумеваних правоугаоника. На нивоу једног слоја користе се исти подразумевани правоугаоници за сваку ћелију и они су центрирани у тој ћелији, али различити слојеви

користе различите подразумеване правоуганике услед разлике у резолуцији. Предикције се могу класификовати као позитивна и негативна поклапања. Ако одговарајући подразумевани правоугаоник има IoU већи од 0.5 са ручно означеним правоугаоником (енг. ground truth, GT), поклапање се сматра позитивним, а у супротном негативним. На слици 17 у левом делу приказани су GT, а у средини имамо два подразумевана правоугаоника поклопљена са мачком и један са псом (десно), који се сматрају позитивним преклапањима, а остали правоугаоници се сматрају негативним. На слици 17 је приказан и пример како SSD детектује објекте различитих величина и односа ширина/висина. Пас одговара једном подразумеваном правоугаонику (црвеном) у слоју са 4×4 мапом фичера, али ниједном подразумевао правоугаонику у мапи фичера веће резолуције (8×8). Мачка је мања и може се детектовати само коришћењем слоја са 8×8 мапом фичера, у 2 подразумевана правоугаоника (плави). На крају се користи потискивање не-максимума како би се уклониле предикције дубликати. Предикције се сортирају по поузданости и задржава се најбољих 200 по слици [21].

Овај модел по перформансама надмашује Faster R-CNN модел, који је до тог тренутка сматран најбољим, достижући 74.3% mAP на PASCAL VOC 2007 тесту и 59 FPS. Један од недостатака SSD-а је што даје слабије резултате за мале објекте (они могу бити детектовани само у слојевима највеће резолуције који садрже мање значајне фичере). Тачност детекције модела расте са повећањем броја подразумеваних правоугаоника, али по цену брзине.

3.2. Препознавање лица

Када је реч о обради слика лица прво треба дефинисати појам детекције лица зато што је то процес који најчешће претходи било којој обради лица. Детекција лица је рачунарска техника која се користи за проналажење људског лица на дигиталним сликама, а односи се на испитивање да ли се на слици налази лице, где је лоцирано и које је величине [22]. Као најпознатији алгоритми који се користе за детекцију лица могу се издвојити Viola-Jones и HOG заједно са Linear SVM (описани у претходном поглављу), Eigenfaces и методе засноване на неуронским мрежама. Алгоритми се генерално разликују по понашању при екстремним условима – као што су лоше осветљење, различити положаји лица, изрази лица, лица веома мале или велике резолуције, прекривеност лица (нпр. наочаре, коса, брада), итд. Након детекције лица могу се издвојити карактеристичне тачке лица (енг. facial landmarks) које означавају истакнуте делове лица попут очију, обрва, носа, усана и образа. На пример, карактеристичне тачке се успешно примењују за поравнање лица, препознавање лица, одређивање положаја главе...

Препознавање или идентификација лица је процес утврђивања коме одређено лице са слике припада, односно, упоређивање неког лица са познатим лицима из базе података у циљу проналажења поклапања [23]. Ово је процес који се у великој мери користи у свакодневном животу. Не мора се у свим применама радити упоређивање са лицима из базе података јер је некада потребно само дозволити приступ одређеној особи, а осталима забранити (као на пример код откључавања телефона). Поред 2D система, постоје и 3D системи за препознавање лица који укључују додатне информације о облику лица у простору и они могу дати боље резултате него 2D системи. У циљу побољшања резултата препознавања лица често се врши поравнање лица, при чему неке методе посматрају 3D модел лица, а једноставније методе се ослањају само на карактеристичне тачке лица (конкретно, на тачке које одређују очи).

Код система за препознавање лица може бити проблема уколико неко злонамерно покуша да приступи систему представљајући се као неко други. Може се десити да неко подметне слику или видео друге особе и, уколико нема провере да ли је лице на камери реално („живо“), систем подметнута лица може препознати као валидна. Постоје различити приступи овом проблему, а неки од њих су анализа текстура, анализа фреквенци (Фуријеов домен лица), анализа променљивих величина између узастопних фрејмова, алгоритми базирани на хеуристикама (покрети очију, усана, трептање), алгоритми оптичког тока (разлике у оптичким токовима генерисаним од стране 3D и 2D равни), посматрање 3D облика лица или комбинација наведених приступа [24].

Временом, најпре су били развијени системи који користе геометрију лица за идентификацију (позиција и величина очију, носа, образа и браде), затим системи који користе алгоритме машинског учења (издвајање фичера и тренирање класификатора), а у скорије време за препознавање лица све више се користе алгоритми дубоког учења. Један од историјски најзначајнијих алгоритама је Eigenfaces, где је коришћена техника линеарне алгебре за редукцију димензија која се зове Principal Component Analysis (PCA). Затим је представљена метода Fisherfaces која користи Linear Discriminant Analysis (LDA), као и метода Local Binary Patterns (LBP), базирана на фичерима, која се и данас користи у многим апликацијама. Када је у питању дубоко учење, постоје специјалне архитектуре које се зову сијамске мреже (енг. siamese networks). Ове архитектуре садрже две или више идентичних неуронских мрежа којима се прослеђују улази за које треба одредити колико су слични. FaceNet је један од најпопуларнијих модела дубоког учења који се користи за препознавање лица [25].

3.2.1. Eigenfaces

У раду *Face Recognition Using Eigenfaces* [26] из 1991. године, Turk и Pentland представили су Eigenfaces алгоритам за детекцију и препознавање лица, који се заснива на редукцији димензија простора слика лица коришћењем анализе главних компоненти (PCA). За представљање лица користе се својствени вектори (енг. eigen vectors), односно, својствена лица (енг. eigenfaces).

Алгоритам најпре обрађује скуп од M тренинг слика лица које су сиве, исте величине и поравнате тако да се кључни делови лица што више поклапају. Посматране [27] су слике димензија $N \times N$ (у општем случају могу бити произвољне величине) које се могу линеаризовати и представити као вектор димензија N^2 , односно, као тачка у простору са N^2 димензија. Израчунава се средње лице које је представљено вектором дужине N^2 , чији су елементи просечна вредност одговарајућих елемената вектора тренинг слика, након чега се свака тренинг слика представља као разлика оригиналне слике и слике средњег лица. Овиме се заправо помера средња тачка (средње лице) у координатни почетак како би се лакше израчунала коваријанса. Варијанса представља одступање вредности тачака од средње вредности и рачуна се за сваку од димензија, а коваријанса представља меру повезаности димензија. Матрица коваријансе има димензије које одговарају броју димензија простора са којим радимо (у нашем случају $N^2 \times N^2$) и може се израчунати као $C = AA^T$, при чему је матрица A добијена смештањем вектора тренинг слика у колоне. У пресеку неке врсте и колоне матрице коваријансе налазиће се вредност коваријансе димензија које су везане за ту врсту и колону. Циљ примене PCA је пронаћи принципалне компоненте које најбоље описују дистрибуцију слика лица у тренинг скупу, односно *eigenvector*-е матрице коваријансе која одговара скупу тренинг слика. Eigenvector је вектор димензија N^2 и може се посматрати као замућено лице које називамо „eigenface“ или „својствено лице“. Рачунање матрице

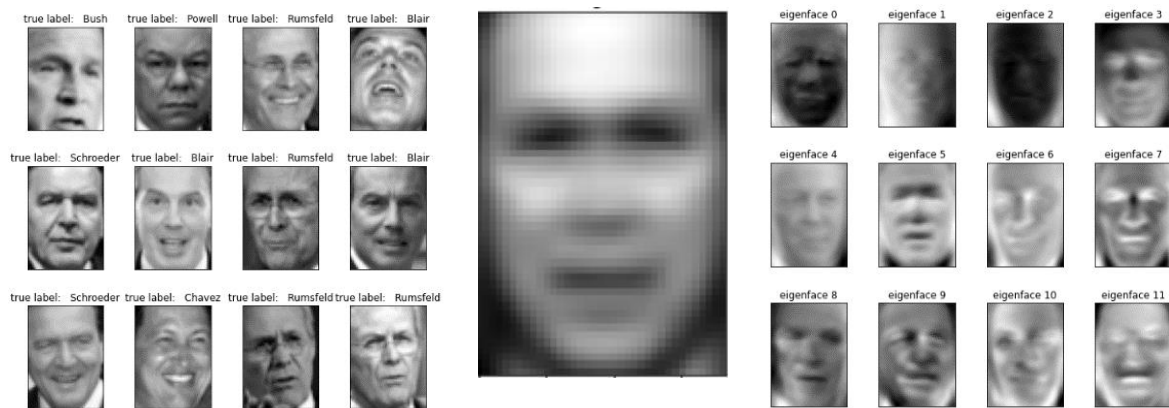
коваријансе може се ефикасније обавити множењем A^T и A , чиме се добија матрица C' димензија $M \times M$, при чему важе следеће релације:

$$A^T A v_i = \lambda_i v_i \quad (4)$$

$$A A^T A v_i = \lambda_i A v_i \quad (5)$$

$$C u_i = \lambda_i u_i \quad (6)$$

где су λ_i својствене вредности (eigenvalues) обе матрице, v_i својствени вектори матрице C' , а u_i својствени вектори матрице C . На основу израчунатих вектора v_i могу се добити вектори u_i коришћењем једнакости $u_i = A v_i$. Свако лице из тренинг скупа може бити представљено као линеарна комбинација K својствених лица која најбоље репрезентују варијације међу лицима и која описују K -димензиони простор који називамо „простор лица“, при чему је K много мање од иницијалне димензионалности простора слика лица. Тренинг слике се могу представити вектором тежина које се јављају у линеарној комбинацији. На слици 18 лево су приказана тренинг лица, у средини је представљено средње лице, а десно је приказано првих 12 eigenface репрезентација. Светлији региони одговарају већим варијацијама, а тамнији мањим.



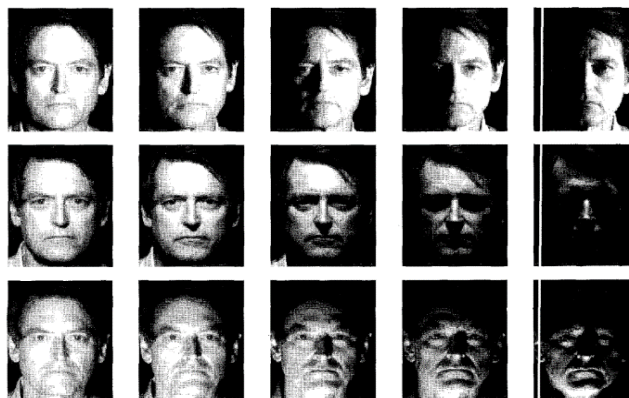
Слика 18: Тренинг лица (лево), средње лице (середина), eigenfaces (десно) [27]

Када се анализира нова слика лица, она се модификује слично као тренинг слике тако што се од ње се одузима средње лице, а затим се пројектује на K -димензиони простор као линеарна комбинација својствених лица. За потребе детекције лица, из K -димензионог простора се реконструише слика (тако што се средњем лицу дода сума производа својствених лица и одговарајућих тежина придружених пројектованој слици) и пореди са оригиналном сликом. У случају да је та разлика већа од граничне вредности, на слици нема лица, а у супротном је лице детектовано. Уколико је циљ препознавање лица, онда се добијени K -димензиони вектор пореди (рачуна се еуклидско растојање) са векторима добијеним пројекцијом тренинг слика. Што је мање растојање, лица су сличнија. Лице се идентификује као лице са којим има најмање растојање [27].

Eigenfaces је био један од првих значајнијих алгорита намењених препознавању лица. Предности овог алгорита су што није рачунски захтеван, брзо се извршава и не захтева знање о карактеристикама лица. Један од недостатака овог алгорита је што захтева поравнање слика лица приликом тренирања и препознавања (зато што се ради на нивоу пиксела, тако да је потребно да се лица скоро савршено поклапају). Такође, захтева фронтална лица и осетљив је на промене у положају и осветљењу. На ефикасност препознавања могу утицати величина лица, експресија лица и позадина која се налази око лица.

3.2.2. Fisherfaces

Fisherfaces метод представљен је 1997. године у раду **Belhumeur** и сар. *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection* [28]. Уз Eigenfaces, ово је један од најпознатијих алгоритама за препознавање лица. И он се заснива на редукцији димензија, али користи линеарну дискриминантну анализу (LDA). Сваки пиксел на слици посматран је као тачка у високодимензионом простору и слика је линеарно пројектована у нискодимензиони потпростор који није осетљив на промене у осветљењу и експресији лица.



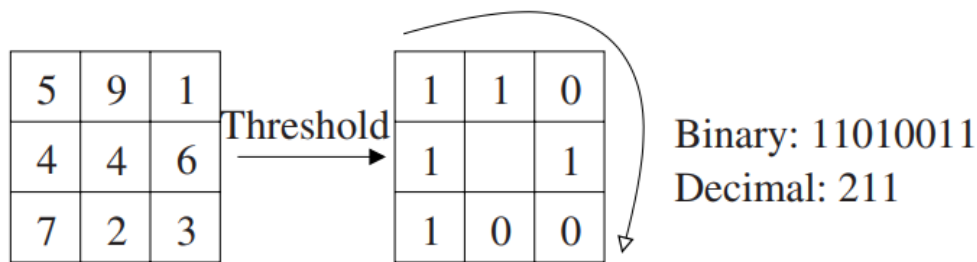
Слика 19: Сlike лица при различитом осветљењу [28]

Eigenfaces користи пројекције које максимизују разлике између лица у целом скупу слика, што не даје добре резултате при препознавању када постоје промене у осветљењу јер су веће разлике између слика истог лица при различитом осветљењу него између слика лица различитих особа, што се може уочити на слици 19. Fisherfaces користи пројекцију која ће максимизовати однос расподеле између класа и расподеле унутар класа. Тежи се максимизацији удаљености лица различитих класа, а минимизацији удаљености унутар сваке класе. Рачунају се матрице коваријансе између класа и унутар класа, а затим својствени вектори и својствене вредности. Бирају се вектори са највећом својственом вредношћу. Као код Eigenfaces алгорита, тренинг слике се пројектују у потпростор слика, улазна слика се такође пројектује у овај потпростор и упоређују се растојања [28].

Метод је дао боље резултате од Eigenfaces алгорита и није осетљив на велике варијације у осветљењу (које се односе на промену у интензитету, али и смеру и броју светлосних извора) и експресији лица. Један од недостатака алгорита је што не подржава промене у пози и, слично Eigenfaces алгоритму, захтева да слике лица буду поравнате. Налажење пројекција је рачунски захтевније него код Eigenfaces метода, али је Fisherfaces метод поузданији.

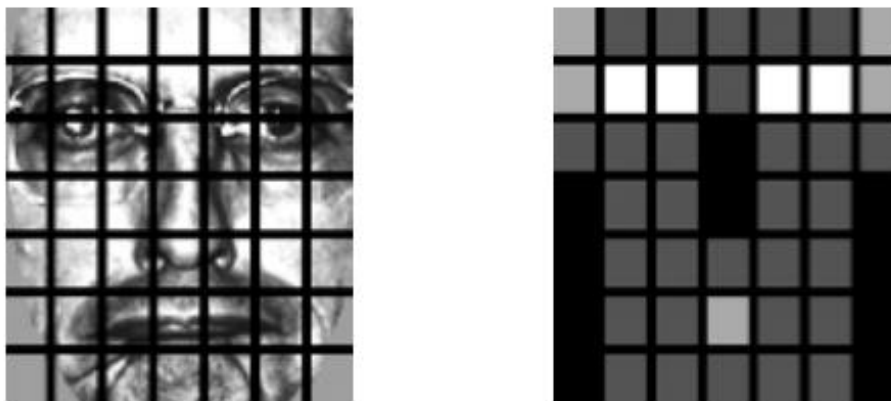
3.2.3. Local Binary Patterns

Алгоритам који користи Local Binary Patterns (LBP) за препознавање лица приказан је у раду **Ahonen-a** и сар. *Face Recognition with Local Binary Patterns* [29], објављеном 2006. године. Метод узима у обзир информације о облику и текстури приликом представљања слике. LBP оператор сваки пиксел на слици пореди са околним пикселима и резултат посматра као бинарну вредност. Пример примене оператора дат је на слици 20. Вредностима већим од 4 придружена је јединица, а мањим од 4 придружена је нула, након чега се добијена осмобитна вредност преводи у децимални број. Затим се хистограм ових вредности користи као дескриптор текстуре.



Слика 20: LBP оператор [29]

Први корак у алгоритму је превођење слике у сиву и подела на матрицу од 7×7 једнаких ћелија. Затим се за сваку од ћелија израчуна LBP хистограм фичера. Ако се посматра 8 суседних пиксела, биће $2^8 = 256$ могућих вредности, те је хистограм дужине 256. Рачунањем хистограма за сваку ћелију заправо енкодирамо просторне информације као што су очи, нос, уста... Неки региони слике носе више информација, неки мање, тако да се уводе тежине које се придружују ћелијама. На слици 21 лево је приказано лице подељено на ћелије, а десно тежинска шема. Бели квадрати (очи) имају тежину 4 (њихови хистограми су помножени са 4), светлосиви тежину 2 (уста и уши), тамносиви 1 (унутрашњи образи и чело), док црни имају тежину 0 (нос и спољашњи део образа). Вредности тежина су експериментално утврђене. Тежински хистограми се надовезују и формирају јединствени хистограм фичера који репрезентује слику лица. Препознавање лица се врши упоређивањем растојања. Улазно лице се обрађује као и тренинг лица (издвајају се LBP, додају им се тежине, надовезују се) и затим се примењује k-NN (k-Nearest Neighbors) алгоритам ($k=1$) са χ^2 растојањима, како би се нашло најближе лице из тренинг скупа [29].



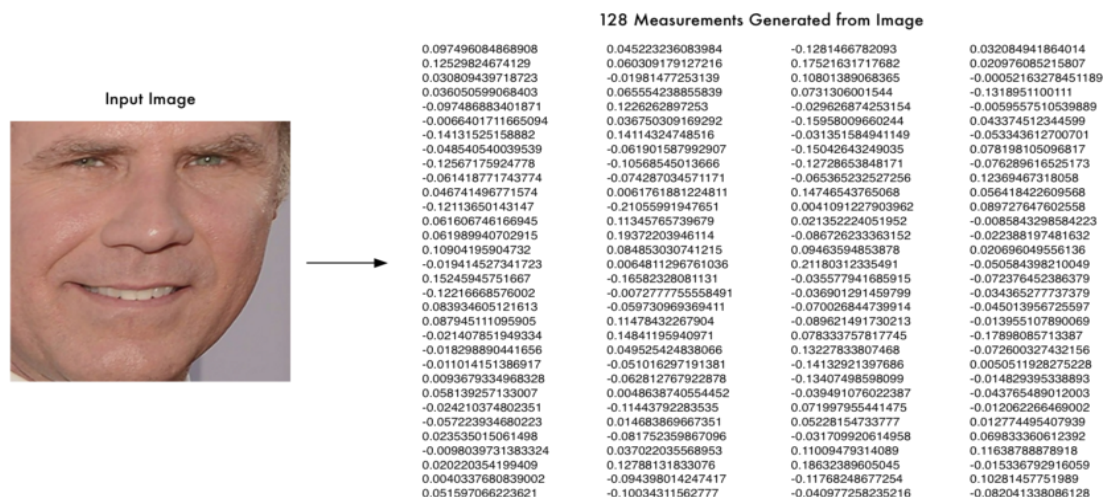
Слика 21: Слика лица подељена на ћелије (лево), тежинска шема за одговарајуће ћелије (десно) [29]

LBP представља локалну репрезентацију текстуре. Овај алгоритам може да се ажурира приликом додавања новог лица у систем, док многи алгоритми захтевају да сва лица која ће се идентификовати буду присутна у време тренирања. LBP алгоритам је отпорнији на шумове јер не ради директно са интензитетима пиксела и углавном даје боље резултате него Eigenfaces. Ефикасан је и омогућава веома брзо издвајање фичера.

3.2.4. Дубоко учење

Дубоке конволуционе неуронске мреже могу се користити за препознавање лица са великом тачношћу. Један од најпознатијих система, назван „FaceNet“, представљен је у раду **Schroff**-а и сар. *FaceNet: A Unified Embedding for Face Recognition and Clustering* [30]. Мрежа је тренирана да генерише вектор од 128 елемената за свако лице који се може користити при упоређивању лица. Мери сличности два лица директно одговарају растојања између њихових вектора. На широко коришћеном скупу слика Labeled Faces in the Wild (LFW), овај систем постиже тачност од 99.63%.

При препознавању лица, на почетку треба детектовати лице, издвојити карактеристичне тачке и поравнати га коришћењем афиних трансформација тако да очи и уста буду увек на истим позицијама на слици, како би било олакшано упоређивање. За разлику од неуронских мрежа за детекцију објеката које се тренирају да би препознале објекат на слици, ове неуронске мреже се тренирају да генеришу 128 мера (низ реалних бројева) које карактеришу свако лице (слика 22). Идеја да се компликовани подаци попут оних које носи једна слика представе преко 128 бројева је била веома значајна за машинско учење. Ови бројеви називају се „embeddings“. Тренирање мреже се обавља тако што се користе 3 различите слике (енг. triplets), при чему две припадају истој особи, а трећа некој другој особи. Тежине мреже се подешавају тако да слике исте особе имају врло сличне векторе, а слике те особе и неке друге што различитије. Овај поступак се понавља милионима пута за милионе слика да би мрежа научила да поуздано генерише мере за сваку особу. Када се мрежа једном истренира може се користити за генерисање вектора за било које лице, па и потпуно непознато. Да би се непознато лице препознато треба наћи у бази познатих лица оно које има најближе мере (за овај део може се користити нека од метода за класификацију) [31].

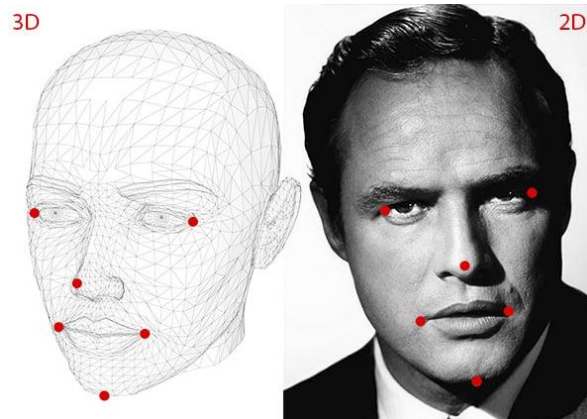


Слика 22: Пример 128 мера лица које генерише неуронска мрежа [31]

Дубоко учење се данас врло често примењује у домену препознавању лица и може се рећи да је у потпуности заменило традиционалне алгоритме. Недостатком овог метода може се сматрати потреба за тренирањем мреже огромним бројем слика (које садрже довољно варијација) како би могла добро да генерализује. Међутим, данас су доступни велики припремљени скупови података који се могу директно користити, као и већ истрениране мреже.

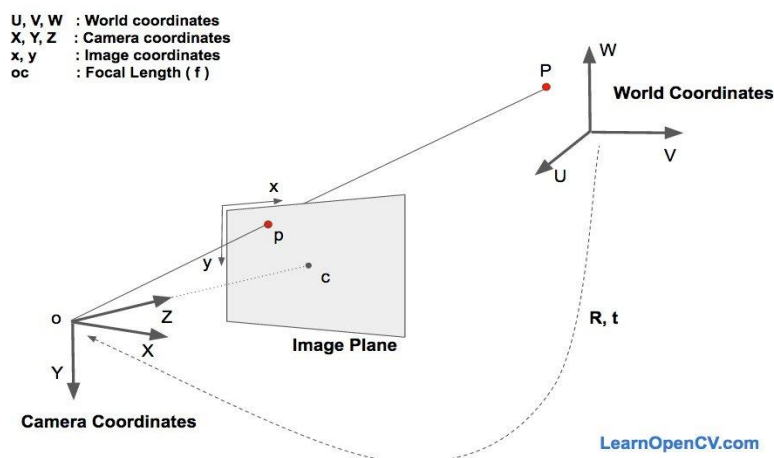
3.3. Одређивање положаја главе

Поза неког објекта се посматра као његова релативна оријентација у односу на камеру. Проблем естимације позе се често назива Perspective-n-Point проблем или PnP. Положај се одређује када имамо калибрисану камеру и знамо локације n 3D тачака које одговарају 2D пројекцијама на слици. На слици 23 је дат пример 6 тачака у 3D простору и одговарајућих 2D тачака које се могу користити за одређивање положаја главе [32].



Слика 23: 6 карактеристичних 3D тачака лица и одговарајућих 2D тачака [32]

Објект се може кретати у односу на камеру применом транслације или ротације, а то кретање можемо посматрати и као кретање камере у односу на објект. Транслацијом се назива померање камере са њене тренутне 3D позиције (X, Y, Z) на нову 3D локацију (X', Y', Z') и она се може представити вектором $t = (X' - X, Y' - Y, Z' - Z)$. Ротацијом се назива ротација камере око X, Y и Z осе и може се представити коришћењем Ојлерових углова, 3×3 ротационе матрице или смером ротације и углом. За одређивање позе главе потребно знати 2D координате (x, y) неколико тачака на слици (врх носа, врх браде, углови очију и углови усана), 3D координате истих тачака, као и унутрашње параметре камере (фокалну дужину камере, оптички центар слике и параметре радијалне дисторзије). Оптички центар слике се може апроксимирати центром слике, фокална дужина ширином слике и може се претпоставити да радијална дисторзија не постоји.



Слика 24: Координатни системи [32]

Посматрају се 3 координатна система приказана на слици 24: светски координатни систем, координатни систем камере и координатни систем слике. 3D координате тачака представљене у светским координатама, уз познату ротацију и трансформацију, могу се трансформисати у 3D тачке у координатама камере. Ове тачке могу бити пројектоване на раван слике коришћењем унутрашњих параметара камере, чиме се добијају тачке у координатном систему слике. Разматрају се једначине које дају пројекцију р 3D тачке P, која је на позицији (U, V, W) у светском координатном систему, на раван слике. Ако је позната ротациона матрица R (3×3) и транслациони вектор t (3×1), могу се израчунати (X, Y, Z) координате тачке P у координатном систему камере:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R \begin{bmatrix} U \\ V \\ W \end{bmatrix} + t = [R \mid t] \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix}, \quad (7)$$

односно:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix}. \quad (8)$$

Ако постоји довољан број парова (X, Y, Z) и (U, V, W), може се решити систем линеарних једначина који ће дати непознате r_{ij} и (t_x, t_y, t_z) . При одређивању положаја главе су познате тачке у 3D моделу (U, V, W) и 2D тачке (x, y), али не и (X, Y, Z). У одсуству радијалне дисторзије координате (x, y) тачке p дате су једначином:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (9)$$

где су f_x и f_y фокалне дужине у x и у смеровима, а (c_x, c_y) је оптички центар. Непознати фактор скалирања s постоји у једначини услед чињенице да није позната дубина слике. Спајањем било које 3D тачке P са центром камере, добиће се пројекција тачке P – тачка p у којој одговарајући зрак пресеца раван слике. Све тачке дуж једног зрака имаће исту пројекцију на раван слике, тако да се коришћењем претходне једначине може добити само (X, Y, Z) са скалирањем s. Сада једначина (8) изгледа овако:

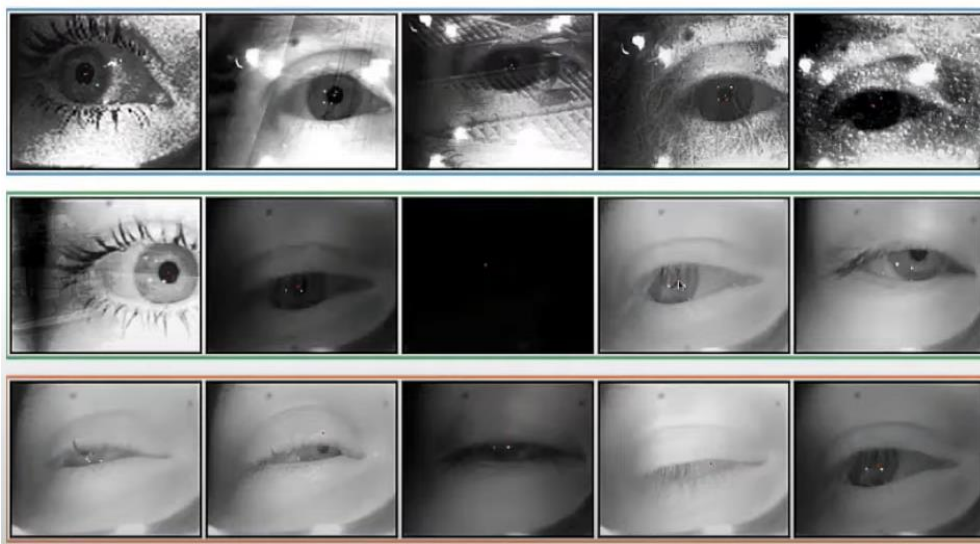
$$s \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \quad (10)$$

и може се решити коришћењем метода који се зове директна линеарна трансформација (DLT). Овај метод се може користити кад год постоји проблем где је једначина скоро линеарна, али је помножена непознатим фактором. Из једначина (8) и (9) се види да се, уз познавање R и t, може наћи тачка p на слици за сваку 3D тачку P (пројектовањем 3D тачака на 2D слику). R и t се могу добити DLT методом. Како су познате 2D карактеристичне тачке лица, може се посматрати растојање између пројектованих тачака и тих тачака и израчунати грешка пројекције. Коришћењем метода који се зове Levenberg-Marquardt оптимизација, резултат се може побољшати (итеративном променом вредности R и t тако да се грешка пројекције смањује) [32].

3.4. Праћење погледа

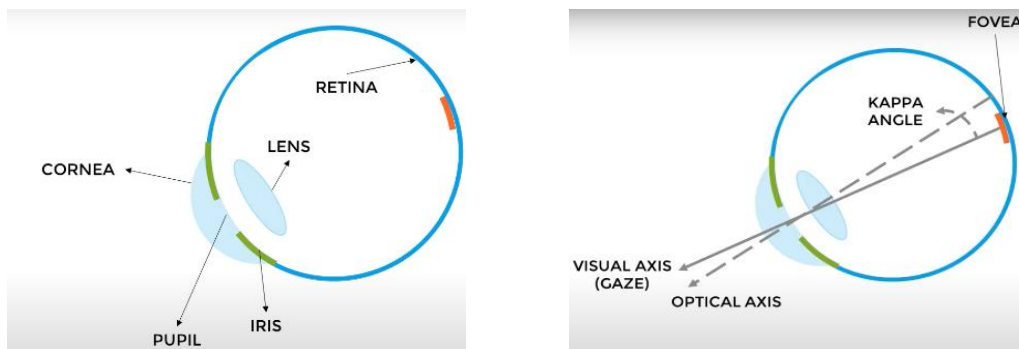
Праћење покрета очију је добро познат проблем у рачунарском виду и постоје решења која се успешно примењују за различите намене попут управљања показивачем на екрану очима, помоћи људима са одређеним видовима инвалидитета, анализе пажње.

Постоје два случаја праћења погледа: када треба одредити тачку на екрану у коју особа гледа и када треба одредити тачку у сцени у коју особа гледа. У оба случаја користи се систем са две камере [32]. У првом случају једна камера у суштини представља екран, а друга снима лице особе и најчешће је закачена за екран. У другом случају особа носи специјалне наочаре које имају камеру у средини (која је усмерена ка сцени) и другу камеру која је закачена за наочаре и усмерена ка очима (најчешће је ка очима усмерено и инфрацрвено светло које се користи да би се добио бољи контраст на слици, а при томе не утиче на очи јер га људско око не може спознати). Један од најважнијих корака у одређивању смера гледања је детекција зенице. Проблеми који се могу јавити при детектовању зенице су следећи: да око може бити снимано из различитих углова; особа може носити наочаре; очи јој могу бити делимично затворене; трепавице могу делимично прекрити око; постоје варијације у облику ока код људи; квалитет слике и осветљење могу бити лоши (слика 25).



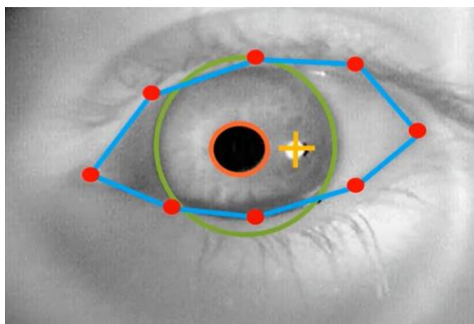
Слика 25: Варијације у сликама очију [33]

Алгоритми за детекцију зенице могу се засновати на класичним техникама, као и на конволуционим неуронским мрежама. Да би се одредио положај зенице потребно је осврнути се на грађу људског ока приказану на слици 26. Светлост пролази кроз отвор који се зове зеница (енг. pupil), коју контролише дужица (енг. iris), па се прелама кроз сочиво (lens), после чега се слика пројектује на мрежњачу (енг. retina). Око зенице налази се рожњача (енг. cornea). Линија која повезује центар зенице и центар сочива назива се оптичка оса. Место на мрежњачи где је слика најоштрија зове се *fovea* и налази се у центру жуте мрље. Линија која спаја ову тачку и центар сочива назива се визуелна оса и она представља правац погледа. Угао између ове 2 осе се назива *kappa* угао и најчешће је око 5 степени. Визуелну осу је могуће одредити на основу оптичке, пре чега је потребно урадити калибрацију за сваку особу. Калибрација подразумева да се корисник фокусира на одређену тачку, затим се одреди тачка у коју гледа и нађе разлика између локације добијене тачке и стварне тачке. Та информација се користи да се побољша наредна процена.



Слика 26: Грађа ока (лево), оптичка и визуелна оса (десно) [33]

Када светлосни зрак погоди рожњачу формира се одсјај у оку у виду белог кружића. Неке од карактеристика ока која могу бити значајне за одређивање тачке погледа су означене на слици 27. Ради се о центру и полупречнику зенице, локацији одсјаја рожњаче, центру и полупречнику дужице и контури (полигону) ока.



Слика 27: Значајне карактеристике ока [33]

Алгоритми за праћење погледа могу се поделити у неколико група: алгоритми засновани на 3D моделу, алгоритми засновани на регресији, Cros-Ratio алгоритми и алгоритми засновани на изгледу. Алгоритми засновани на 3D моделу главе и ока могу користити и RGBD камере за одређивање 3D параметара. Око и рожњача се могу моделовати као сфере, а зеница и дужица као концентрични кругови. Овај модел је најпрецизнији, добро се понаша при покретима главе и различитим светлосним условима, али је и најкомплекснији и захтева калибрацију и доста хардвера. Методи засновани на 2D регресији мапирају 2D фичере на поглед без експлицитног моделовања геометрије. Неки од оваквих метода су креирање полинома на основу фичера, Support Vector Regression (SVR) и неуронске мреже. Код Cros-Ratio метода се постављају четири светла у угловима екрана, а камера и извор светлости су на истој локацији тако да се добије светла зеница. Уз познавање центра зенице, рефлексije ових светала у оку и претпоставку да је рожњача равна површина, може се одредити тачка погледа. Алгоритми засновани на изгледу имају за циљ рекреирање региона ока коришћењем 3D морфолошког модела ока. Овај метод је најмање прецизан, али и има најмање хардверске захтеве [33].

У апликацијама за надгледање полагања тестова може се детектовати трептање и одређивати смер гледања особе која ради тест. Тачка екрана у коју особа гледа се може пратити уколико је потребно одрадити анализу пажње, али за утврђивање регуларности полагања то није неопходно, већ се може проверити само да ли гледа са стране, тј. ван екрана. Праћење треба обављати у реалном времену на видеу са стандардне веб камере јер је претпоставка да ученици у кућним условима немају никакву додатну хардверску опрему и не могу задовољити захтеве напреднијих система.

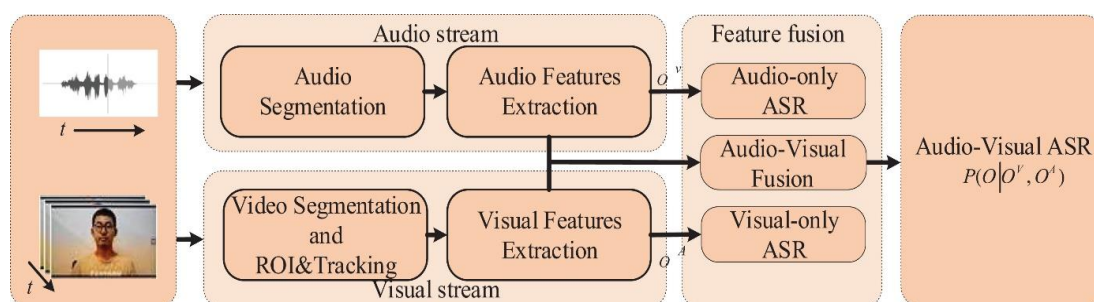
3.5. Детекција говора

Већина онлајн тестова не захтева усмене одговоре и причање током теста је забрањено, тако да у системима за надгледање полагања може бити довољно проверавати да ли особа прича или не (енг. voice activity detection). Додатно, може се вршити препознавање говора уколико желимо да у извештају наведемо шта је особа изговарала (енг. speech recognition).

Алгоритми за детекцију говора у аудио сигналу најчешће прво редукују шум, затим издвајају одређене карактеристике сигнала из сваког сегмента сигнала и након тога примењују класификационо правило којим се сегмент класификује као сегмент са говором или као сегмент без говора. Карактеристике које се могу користити су спектрални нагиб, коефицијенти корелације, кепстрални коефицијенти итд. Већина мера се заснива на информацијама о фреквенцијама присутним у сигналу. Детекција говора је изазован задатак због постојања позадинске буке која може значајно утицати на звучни сигнал [34].

Већина метода за препознавање говора користи скривени Марковљев модел (енг. Hidden Markov Model, HMM) као основу. У овом моделу претпоставка је да говорни сигнал, када се посматра у кратком временском интервалу (10 ms), може бити апроксимиран стационарним процесом (у коме се статистичке особине не мењају током времена). Други приступ је коришћење неуронских мрежа које се најчешће употребљавају када постоји лош квалитет сигнала, бука или више говорника. У многим модерним системима се користи хибридни модел: неуронске мреже се користе да поједноставе говорни сигнал пре него што се изврши HMM препознавање. Посебно су корисне рекурентне неуронске мреже (енг. Recurrent Neural Networks, RNNs) које су широко коришћене у обради природних језика [35].

Када је реч о видео снимцима, могуће је добити прихватљиве резултате и посматрањем само видео података. Технике рачунарског вида се могу употребити за детекцију причања и читање речи са усана (овај процес се назива „визуелно препознавање говора“). Традиционалне методе могу посматрати карактеристичне тачке лица или посматрати карактеристике усана на нивоу пиксела, а у последње време се све више примењују методе засноване на 3D конволуционим мрежама. Међутим, најбољи начин за детекцију и препознавање говора на видео снимцима јесте комбиновање визуелних и аудио података у процесу који се назива аудио-визуелно препознавање говора (енг. Audio Visual Speech Recognition, AVSR). Визуелне информације (покрети усана) су врло значајне када је у великој мери заступљена бука у звучном сигналу. На слици 28 је приказано како изгледа модел једног AVSR система. Постоје два улазна канала (аудио и видео). Паралелно се врше сегментација аудија и издвајање фичера, и сегментација видеа, издвајање региона од интереса (регион уста), праћење покрета усана и издвајање фичера. На крају следи фаза спајања аудио и видео вектора фичера [36].



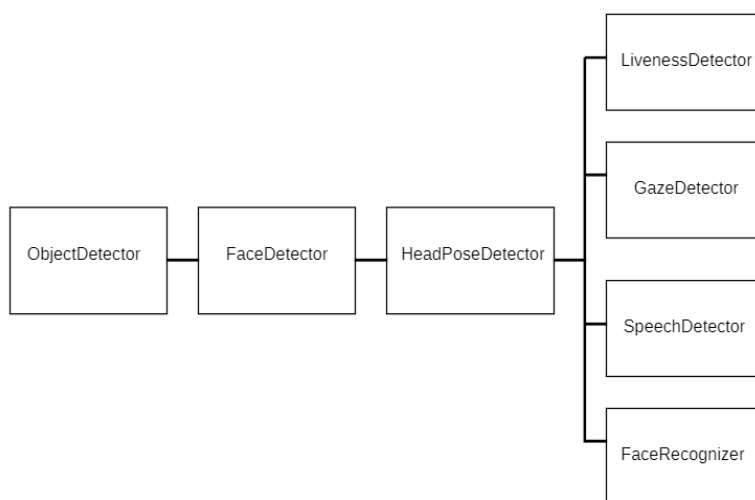
Слика 28: Аудио-визуелни систем за препознавање говора [36]

4. Систем за праћење регуларности полагања тестова употребом камере

За реализацију прототипа система који врши надгледање онлајн полагања тестова употребом камере искоришћен је програмски језик *Python*, а пројекат је имплементиран у развојном окружењу *PyCharm*. Систем не представља платформу преко које се могу полагати тестови, већ само симулира полагање теста укључивањем тајмера. Надгледање полагања се обавља обрадом учитаних фрејмова са камере корисника. За рад апликације потребне су основне информације о ученицима и тестовима, а како ти подаци не би били хардкодирани или чувани у локалном фајл систему, коришћене су *Firebase* услуге³. За обраду фрејмова (слика) највише су коришћене библиотеке *OpenCV* и *Dlib*. Ове две библиотеке су веома популарне и користе се за потребе машинског учења и рачунарског вида.

Кориснички интерфејс апликације је једноставан и интеракција са корисником је имплементирана преко конзоле. Омогућен је избор између неколико акција: додавање/брисање ученика, додавање/брисање теста и покретање теста. Док је тест покренут на снимку са камере који је приказан кориснику исписује се колико је времена остало до краја теста, а у конзоли му се приказују упозорења и обавештења.

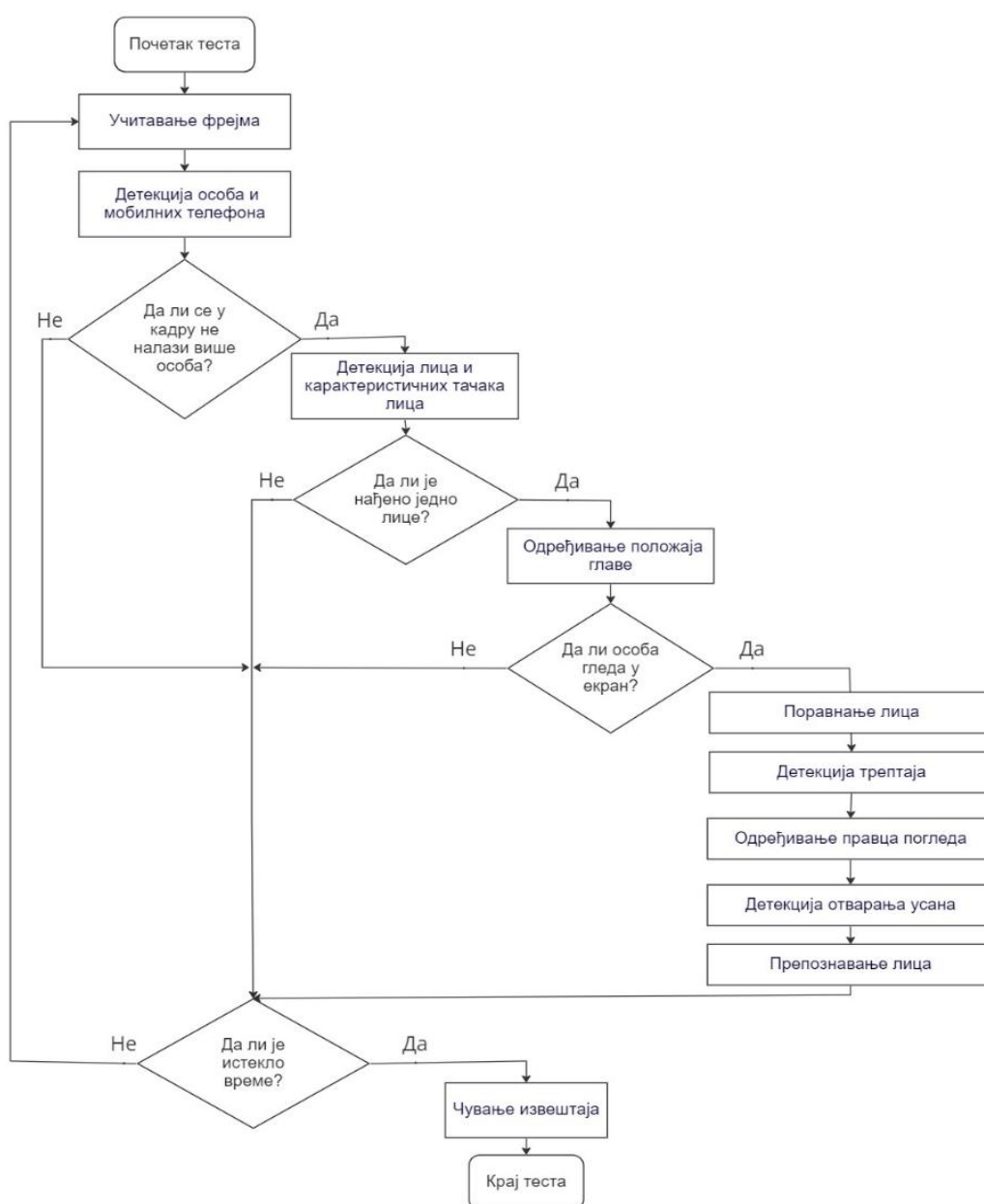
На слици 29 је приказан дијаграм који илуструје компоненте система које се користе за надгледање полагања: *ObjectDetector* – који детектује присуство више особа у кадру и присуство мобилних телефона, *FaceDetector* који детектује лице и издваја карактеристичне тачке лица, *LivenessDetector* – који проверава да ли је детектовано лице „живо“ или је камери подметнута слика, *FaceRecognizer* – који проверава да ли детектовано лице одговара лицу ученика који је покренуо тест, *HeadPoseDetector* и *GazeDetector* – који проверавају да ли су ученикова глава и поглед усмерени ка екрану и *SpeechDetector* – који проверава да ли ученик прича. Свака од компоненти имплементирана је као посебан модул, а из главног фајла, који је улазна тачка апликације, позивају се њихове функције за детекцију и валидацију. Оваква организација кода омогућава да функционалности буду раздвојене и да касније измене у обради фрејмова буду локализоване у оквиру сваког модула.



Слика 29: Компоненте система за надгледање полагања

³ *Firebase* је Гуглова (енг. *Google*) платформа за развој апликација која нуди различите сервисе попут база података, аутентификације и аналитике.

Са базом података се комуницира преко посебног модула названог *Database*. Модел је максимално поједностављен. Подаци који се чувају о ученицима су идентификациони број, име, презиме, једна слика и видео снимци полагања тестова. Претпоставка је да сваки ученик који је убачен у систем има слику попут слика за личне документе на којој му се лице јасно види и гледа у камеру. О тестовима се чувају само информације које се односе на идентификациони број и дужину трајања. Сlike и снимци се чувају у *Firebase Storage*, а остали подаци у *Firestore Database*. За комуникацију са овим сервисима је коришћена библиотека *firebase_admin*. Пре покретања теста, на основу идентификационог броја ученика учитавају се његово име и презиме како би му се приказала одговарајућа поздравна порука и његова слика која се касније користи за препознавање лица, а на основу идентификационог броја теста учитава се информација о трајању теста. Проверава се да ли учитана слика садржи лице, лице се поравнава и укључују се тајмер и камера.



Слика 30: Дијаграм тока обраде фрејмова

Након покретања камере, систем обрађује један по један фрејм. На слици 30 је приказан ток обраде фрејмова. На учитаном фрејму прво се детектују особе и мобилни телефони. Уколико није детектовано више особа у кадру, прелази се на детекцију лица и карактеристичних тачака лица. Уколико је ова детекција успешна, проверава се да ли је глава особе усмерена ка екрану. Ако јесте, следи поравнање лица, након чега се даља обрада врши над поравнаним лицем. Затим се утврђује да ли се десио трептај јер се провера да ли је камери подметнута слика врши бројањем трептаја. Након тога се одређује да ли особа гледа право у екран или са стране и да ли су јој уста отворена јер се праћењем покрета усана закључује да ли особа прича. На крају се врши препознавање лица. Обрада фрејмова се прекида када прође онолико времена на колико је подешено трајање теста (или притиском на тастер „Q“ на тастатури у случају ранијег завршетка теста) и тада се бележи извештај у форми видео фајла који се касније може прегледати ради провере спорних ситуација. Генеришу се два фајла: један који садржи комплетан снимак полагања са означеним спорним ситуацијама, и други, који садржи само спорне ситуације.

Ради лакше обраде фрејмова направљена је посебна класа која представља један фрејм и која има 4 атрибута: слику, вредност тајмера у тренутку читавања фрејма, информацију о валидности фрејма и поруку о нерегуларностима која ће бити исписана на фрејму. Како у једном фрејму може истовремено бити више нерегуларности, сваки од детектора ажурира атрибут поруке дописивањем специфичног текста и може ажурирати атрибут слике доцртавањем додатних информација о детектованој нерегуларности. Сви учитани фрејмови се смештају у бафер главног програма на основу кога се на крају генерише видео запис полагања. Да би се у посебном фајлу издвојиле само спорне ситуације, атрибут валидности сваког фрејма у коме је детектована нека спорна ситуација се поставља на вредност „нетачно“. На крају теста, бафер фрејмова је филтриран по овом атрибуту. Сваки од појединачних детектора има свој „прозор“ бафер у коме чува прозоре фрејмова (прозором се сматра одређени број сукцесивних фрејмова). Прозори фрејмова се посматрају како би се избегле случајне (лажне) детекције. Спорна ситуација неће трајати само један фрејм, већ више узастопних фрејмова. Сви фрејмови који припадају прозору који се сматра нерегуларним биће означени као нерегуларни. У тренутку завршетка теста прекидају се текући прозори свих детектора и позивају се њихове функције за ресетовање. У наставку ће бити детаљније описана свака од коришћених компоненти за надгледање.

4.1. Имплементација система

Прва компонента система, задужена за детекцију објеката, имплементирана је у оквиру модула *ObjectDetector*. При одабиру детектора треба имати у виду конкретан случај коришћења. Један од најбољих и најкоришћенијих модела је YOLO, међутим, оптимизован је за рад на GPU. Ова апликација се извршава на CPU и потребно је да има добре перформансе у реалном времену, тако да је за реализацију детекције објеката изабран MobileSSD модел. У пројекту је коришћена TensorFlow имплементација [37] модела који комбинује MobileNet архитектуру и SSD. MobileNet архитектуре су ефикасније у односу на друге мрежне архитектуре попут VGG или ResNet када су у питању уређаји са ограниченим ресурсима и брже се извршавају, али имају мало мању тачност [38]. Модел је трениран на COCO [39] скупу података са сликама 80 класа објеката који се могу срести у свакодневном животу (људи, аутомобили, животиње, предмети у кући, храна). Овај скуп података садржи око 330 000 слика са 2.5 милиона означених инстанци објеката. Како је циљ овог модула детектовати особе и мобилне

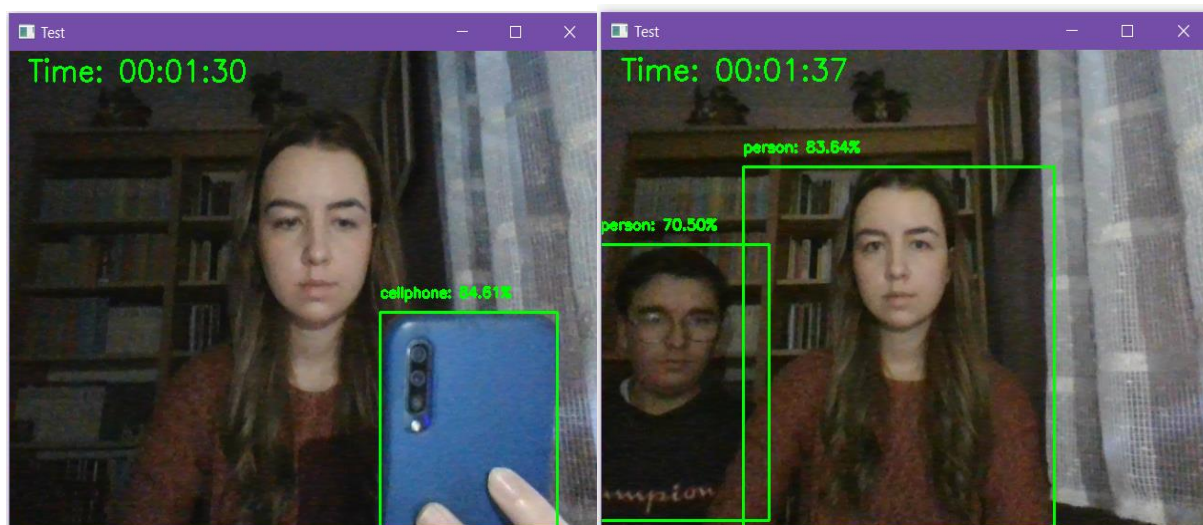
телефоне у кадру, од значаја су биле класе објеката „person“ и „cellphone“. Коришћене су функције из модула за дубоко учење OpenCV библиотеке (доступан од верзије 3.1 и значајно унапређен у верзији 3.3) који омогућава коришћење истренираних мрежа из различитих *framework*-а за дубоко учење као што су нпр. TensorFlow, Torch и Caffe.

На почетку је учитан текстуални фајл који садржи лабеле за сваку класу објеката из СОСО скупа слика. Овај фајл је парсиран тако да се од њега направи низ који садржи називе класа које се могу детектовати. Након тога је уčitана мрежа позивом функције *readNetFromTensorflow*, која као параметре има путању до бинарног фајла који садржи топологију и тежине мреже и путању до додатног конфигурационог фајла. Величина улазне слике (фрејма) је коришћењем функције *blobFromImage* промењена на величину од 300×300 пиксела коју мрежа очекује, слика је нормализована (од сваког пиксела је одузета средња вредност пиксела тренинг слика за сваки од канала), а затим је прослеђена мрежи. Мрежа за сваки детектовани објекат враћа информацију о класи, координатама правоугаоника којим се објекат може обухватити и поузданости детекције. Координате правоугаоника су у опсегу [0, 1], те се множењем тих бројева са оригиналном ширином и висином слике добија предикција за оригиналну слику. Само детекције које имају поузданост већу од 0.6 су разматране и међу њима су издвајане оне које припадају класама од интереса.

За обе класе од интереса направљен је по један бафер прозора, као и помоћни бафер који чува узастопне нерегуларне фрејмове. Тестирањем различитих величина прозора установљено је да је величина прозора од 30 фрејмова довољна да се детектују спорне ситуације. На нивоу сваког прозора се броји колико је било нерегуларних фрејмова и ако тај број прелази једну трећину укупног броја фрејмова у прозору, прозор се сматра нерегуларним. Бафер који чува узастопне нерегуларне фрејмове у једном прозору има улогу да контролише прелаз између прозора. Може се десити да се нерегуларна секвенца налази на прелазу између два прозора, тј. да се један њен део налази у текућем, а други у следећем прозору, и да она не буде урачуната у потпуности (један део упада у прозор који у целини нема довољно нерегуларних фрејмова да би се сматрао проблематичним) или не буде урачуната уопште (ниједан од два прозора нема довољан број нерегуларних фрејмова да се сматрао проблематичним). Из тог разлога се на прелазу из једног у други прозор испитује да ли бафер који садржи узастопне нерегуларне фрејмове није празан. Ако није празан, текући прозор се „скраћује“ за број елемената тог бафера и не отвара се нови прозор док год има сукцесивних нерегуларних фрејмова. Када се наиђе на први валидни фрејм, испитује се да ли у баферу узастопних нерегуларних фрејмова има више елемената од половине величине једног прозора и ако има, ти фрејмови се означавају као нерегуларни. Након тога се прелази на следећи прозор.

Обезбеђена је функција која ће у извештају око детектованог објекта исцртати оквирни правоугаоник и исписати колика је поузданост детекције. На слици 31 у левом делу приказан је пример детекције коришћења мобилног телефона. Треба напоменути да детекцију мобилних телефона „омета“ чињеница да ће најчешће бити држани испод стола, бити прекривени руком или прислоњени уз главу ако особа са неким разговара и у тим случајевима велика је вероватноћа да он неће бити детектован, али ако га особа држи као на слици 31 и покушава да нешто прочита или слика питања са теста, биће детектован. У десном делу слике приказана је ситуација када су детектоване две особе у кадру. Може се десити да модел не детектује особу уколико је мало ближа камери, те из тог разлога ситуација када нема особе у кадру није обрађивана у овом детектору да не би било лажних детекција спорне ситуације. Овај случај је обухваћен у наредном модулу (који детектује лице) тако да се ова два модула допуњују. Ако има више особа у фрејму, он се неће прослеђивати даље, већ ће узроковати ресетовање осталих детектора.

Функција за ресетовање овог детектора проверава да ли је до тог тренутка у баферу нерегуларних фрејмова било довољно елемената да би се обухваћени период сматрао спорним. Ако то није случај, проверава се да ли је обрађено више од две трећине прекинутог прозора и испитује се да ли међу њима има довољно нерегуларних фрејмова.



Слика 31: Детекција мобилног телефона (лево) и детекција 2 особе (десно)

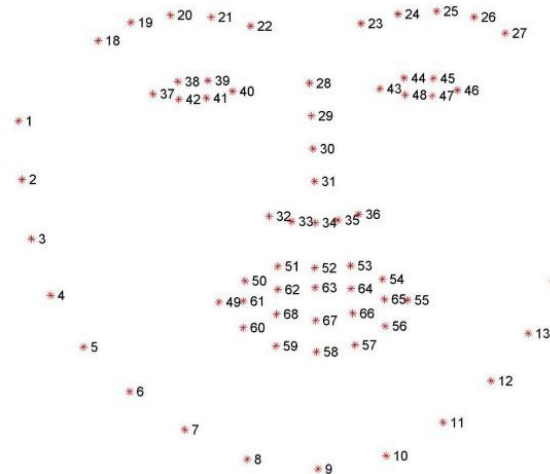
Следећа компонента задужена за детекцију лица и карактеристичних тачака лица имплементирана је у оквиру модула *FaceDetector*. При одабиру детектора лица треба размотрити неколико најзначајнијих детектора: Viola-Jones и DNN детектор из OpenCV библиотеке, као и HOG у комбинацији са линеарним SVM и CNN детектор из Dlib библиотеке. Viola-Jones је најпознатији детектор лица, нема велике захтеве за рачунањем, екстремно је брз и погодан је за извршавање у реалном времену. Са друге стране, подлежан је лажно позитивним детекцијама, захтева ручно подешавање параметара, детектује само фронтална лица и неће радити ако је лице прекривено нечим или није адекватно оријентисано. Треба га користити када је брзина извршавања приоритет и спремни смо да жртвујемо поузданост. HOG алгоритам је поузданији и има стабилнију детекцију од Viola-Jones алгоритма. Његови недостаци су што ради само са фронталним (и благо окренутим) лицима, није толико поуздан као детектори засновани на дубоком учењу, може бити спор јер ради конструкцију клизећег прозора над пирамидом слика, не може детектовати веома мала лица и оквирни правоугаоник некада може да одсече део чела или браде. CNN детектор из Dlib библиотеке има велику тачност, може детектовати делимично прекривена лица, ради добро при различитим оријентацијама лица и светлосним условима, али није погодан за коришћење у реалном времену без GPU убрзања. DNN детектор лица из OpenCV библиотеке је заснован на SSD и ResNet мрежи, брз је и поуздан. Може се извршавати у реалном времену на просечним рачунарима и поузданији је од Viola-Jones и HOG алгоритма, али је мање поуздан од CNN из Dlib библиотеке. Модел ради добро када је лице делимично прекривено, када постоје брзи покрети главе и може детектовати бочна лица, али може се десити да не детектује веома мала лица [40].

За детекцију лица одабран је DNN детектор из OpenCV библиотеке. Коришћена је квантована⁴ TensorFlow верзија [41] модела зато што се извршава за нијансу брже од

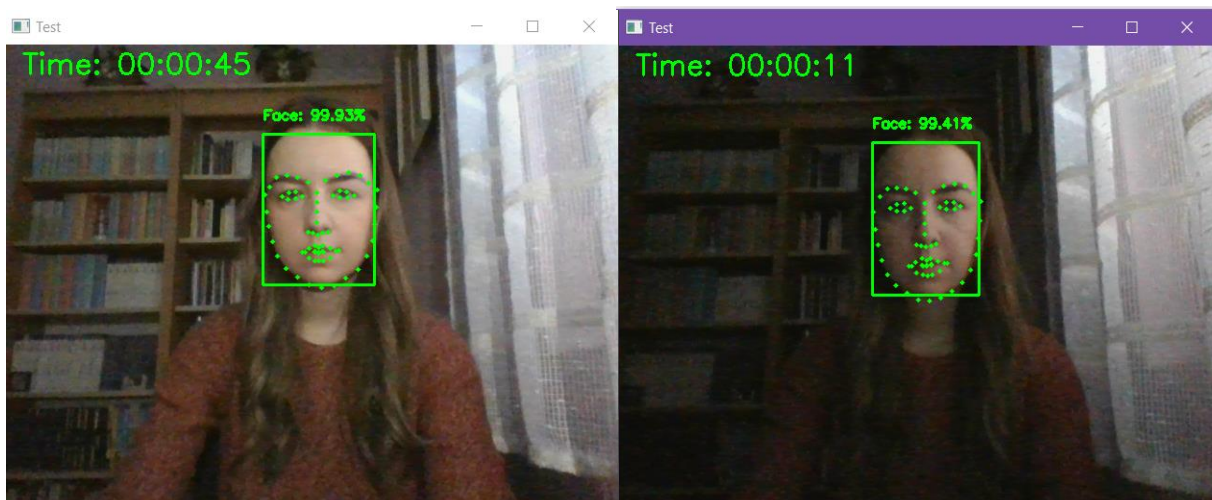
⁴ Квантизација је процес којим се смањује број битова којима се представљају бројеви у неуронској мрежи.

оригиналне. Поступак детекције и обраде детекција је исти као код детекције објеката. Нерегуларним фрејмовима се сматрају фрејмови на којима се не налази тачно једно лице. Овај детектор има еквивалентне функције као детектор особа и када утврди да на слици нема лица или има више од једног лица, ресетоваће се детектори који очекују лице као улаз.

Након детекције лица, детектују се карактеристичне тачке лица које су потребне за даљу обраду лица. Dlib нуди детектор 68 тачака и детектор 5 значајних тачака који је знатно бржи, али препорука је да се он користи у случајевима када су нам потребне само локације носа и очију. Детекција значајних тачака лица на лицу у овом пројекту обављена је коришћењем модела који детектује 68 тачака [42]. Детектују се координате тачака које се мапирају на структуру лица и оне су приказане на слици 32. Имплементирани су и помоћне функције које враћају специфичне тачке лица: тачке левог и десног ока (на слици означене бројевима 37-40 и 43-48), горње и доње усне (на слици означене бројевима 49-68) и 6 карактеристичних тачака за естимацију позе главе (на слици означене бројевима 9, 31, 37, 46, 49 и 55). На слици 33 приказано је детектовано лице заједно са кључним тачкама при различитом осветљењу, где се може приметити да детектор лица ради са веома великом поузданошћу.



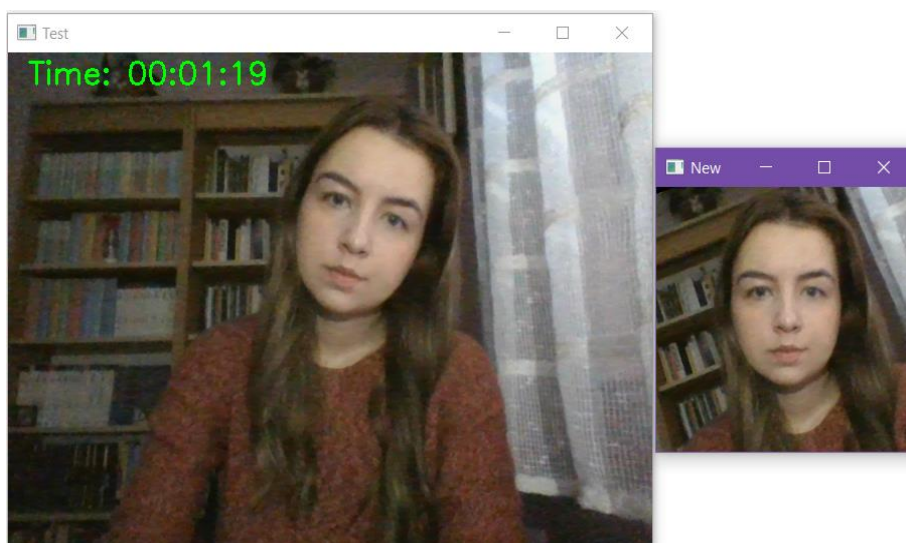
Слика 32: 68 карактеристичних тачака [43]



Слика 33: детектовано лице са карактеристичним тачкама при различитом осветљењу

У овом модулу имплементирана је и функција за поравнање лица. Поравнање лица се ради како би се побољшали алгоритми који ће се примењивати над лицем. У овом пројекту то су препознавање лица, праћење покрета очију и детекција говора. Циљ је да лице буде центрирано на слици, заротирано тако да очи буду на хоризонталној линији и да буде скалирано тако да величина свих лица буде приближно иста. Све слике су смањене на величину од 256×256 пиксела. Слика са поравнатим лицем се враћа позивајућој функцији и даље провере се врше над овом сликом. За постизање поравнања слике коришћена је афина трансформација. Афине трансформације се користе за ротацију, скалирање и транслацију. Ове трансформације могу се објединити у један позив *warpAffine* функције из OpenCV библиотеке.

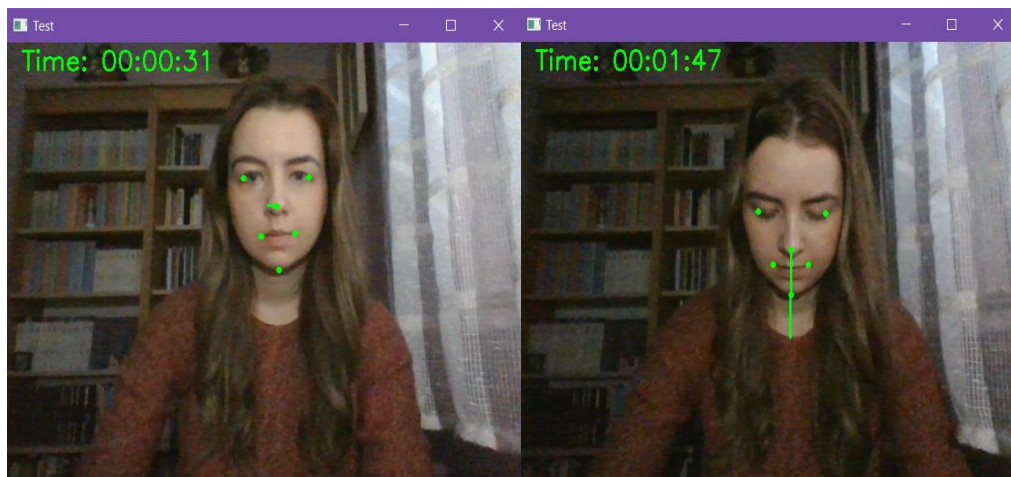
На почетку се задаје жељена (x, y) позиција левог ока. Координате су изражене у процентима ширине и висине оригиналне слике, који су најчешће у опсегу 20-40% (у овом пројекту узето је 40%). Што је мањи проценат, лице ће више бити зумирано. На основу карактеристичних тачака очију израчунавају се центар једног и другог ока, а након тога и угао који линија која их спаја заклапа са хоризонталом. Овај угао је кључан за поравнање слике. Затим је потребно наћи централну тачку која се налази између два ока јер ће се око ње вршити ротација. Након тога израчунава се жељена x позиција десног ока (од 1 се одузме жељена x позиција левог ока, због симетрије), а жељена y позиција десног ока је иста као за лево. Сада можемо израчунати жељену дистанцу између два ока (која ће бити из опсега [0, 1] и помножити је са жељеном ширином коначне слике. Дељењем жељене дистанце са тренутном дистанцом између центара очију можемо израчунати фактор скалирања. За рачунање ротационе матрице M коришћена је OpenCV функција *getRotationMatrix2D* којој су као параметри прослеђени претходно израчунати центар ротације, угао и фактор скалирања. Добијену матрицу потребно је ажурирати, тачније њену транслациону компоненту, како би лице било у оквиру слике и након афине трансформације. Након овога се може применити афина трансформација коришћењем функције *warpAffine* којој се као параметри прослеђују слика, матрица M, жељена ширина и висина излазне слике и опциони параметар којим се специфицира интерполациони алгоритам. За последњи параметар узета је вредност INTER_CUBIC која означава да ће се користити бикубична интерполација која вредност новог пиксела рачуна на основу 16 суседних пиксела [44]. На слици 34 у левом делу приказана је слика где очи особе нису у хоризонталу, а у десном делу обрађена слика где је издвојено поравнато лице.



Слика 34: Оригинални фрејм (лево) и поравнато лице (десно)

Следећа компонента је имплементирана у оквиру модула *HeadPoseDetector* и намењена је детекцији окретања главе. Одређивање положаја главе ученика је имплементирано коришћењем карактеристичних тачака лица и неколико функција из OpenCV библиотеке. Потребно је да знамо да ли је ученикова глава окренута сувише лево, десно, горе или доле. Обрада фрејмова у оквиру прозора и ресетовање су и у овом случају имплементирани као у детектору објеката. Фрејмови у којима ученик не гледа право у екран се не пропуштају наредним детекторима, већ узрокују њихово ресетовање.

Функција која је коришћена за решавање PNP проблема је *solvePnP*. Ова функција као улазне параметре има низ 3D тачака главе (у светском координатама), низ одговарајућих тачака на слици (6 2D тачака лица), унутрашњу (енг. *intrinsic*) матрицу камере, низ коефицијената дисторзије и метод за решавање PNP проблема (може бити селектовано неколико алгоритама, а подразумевани је *SOLVEPNP_ITERATIVE*, који одговара DLT решењу праћеном Levenberg-Marquardt оптимизацијом). Уместо 3D модела лица може се користити генерички 3D модел са тачкама у произвољном (светском) координатном систему: врх носа (0.0, 0.0, 0.0), брада (0.0, -330.0, -65.0), леви угао левог ока (-225.0, 170.0, -135.0), десни угао десног ока (225.0, 170.0, -135.0), леви угао уста (-150.0, -150.0, -125.0), десни угао уста (150.0, -150.0, -125.0). Оптички центар слике се може апроксимирати центром слике, фокална дужина ширином слике и може се претпоставити да радијална дисторзија не постоји. Функција враћа ротациони и транслациони вектор који трансформишу 3D тачке објекта у координате камере [32]. Затим треба издвојити информације о угловима ротације око координатних оса коришћењем *RQDecomp3x3* функције. Ова функција захтева ротациону матрицу, а не ротациони вектор, чију конверзију нам омогућава функција *Rodrigues*. Као координатни почетак се користи врх носа, x-оса је усмерена на лево (из перспективе особе на слици), y-оса на горе, а z-оса право ка екрану. Ротација главе лево-десно одговара ротацији око y-осе, а ротација горе-доле ротацији око x-осе. На слици 35 лево је приказана ситуација када особа гледа право у екран, а десно ситуација када је угао ротације око x-осе већи од граничног.



Слика 35: Пример ситуације када је глава особе усмерена право и на доле, респективно

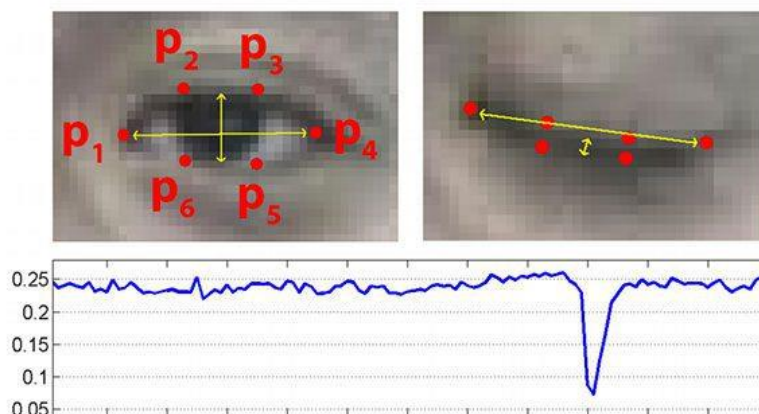
У оквиру модула *LivenessDetector* коришћена је детекција трептања за проверу да ли је лице „живо“, односно, да ли камери није подметнута слика лица. Детектовани су и бројани трептаји ока у одређеном временском интервалу. Ово је један од једноставнијих начина провере и може се преварити подметањем видео снимка. Особа може припремити снимак на коме ради тест и користити га као виртуелну камеру. Систем би овакав начин

варања могао да детектује тако што би повремено захтевао од особе која полаже тест да направи одређени покрет, који подметнути снимак неће моћи да прикаже. У сигурносним системима често се користе термовизијске (инфрацрвене) камере које региструју емитовање топлоте, односно инфрацрвено зрачење које емитују тела, али у условима у којима се треба извршавати овај систем не може се очекивати поседовање оваквих камера.

Како би се детектовао трептај коришћена је метрика EAR (Eye Aspect Ratio). EAR је елегантно решење које врши једноставну калкулацију базирану на односу растојања између одговарајућих карактеристичних тачака очију. Свако око представљено је помоћу 6 тачака, почевши од левог угла ока и крећући се у смеру казаљке на сату. Постоји релација између ширине и висине региона ока:

$$EAR = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2||p_1 - p_4||} \quad (7)$$

где су p_1, \dots, p_6 тачке које одређују око. Ова величина је приближно константна док је око отворено, а нагло опада ка нули када се деси трептај. На графику са слике 36 се може видети EAR кроз време на видео снимку. Вредност је константна (око 0.25), затим нагло опада до нуле, а онда поново расте, што индицира да се десио трептај. Коришћењем ове методе избегавамо технике процесирања слике и ослањамо се на растојања између одговарајућих тачака [45].



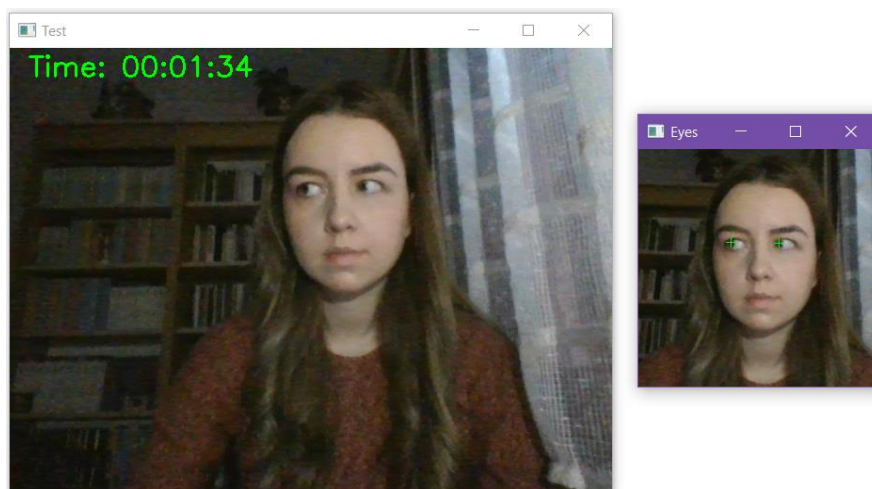
Слика 36: Детекција трептаја [45]

Трептај ока ће трајати минимум два фрејма. Два и више узастопних фрејмова који имају EAR мањи од граничне вредности (0.22) могу се сматрати су трептајем. Људи у просеку трепну између 12 и 22 пута у току једног минута, а како је прозор од једног минута сувише дуг (јер неки тестови могу кратко трајати па можда систем неће стићи да детектује неправилност), посматран је прозор од 30 секунди јер се претпоставља да ниједан тест неће трајати краће од тог времена. Очекује се да ће у том периоду особа трепнути између 6 и 11 пута, али посматран је опсег мало шири од овог због потенцијалних грешака при детекцији. Прозори који имају бројач трептаја ван овог опсега сматрани су нерегуларним. Ресет функција овог детектора враћа на нулу бројаче и празни бафер прозора.

Ученик би током полагања теста требало да гледа право у екран и да не скреће поглед лево, десно, горе или доле. У модулу *GazeDetector* детектују се периоди када постоји дуже гледање ван екрана или када се често на кратко гледа ван екрана, а појединачно краткотрајно гледање са стране се толерише јер то могу бити случајне

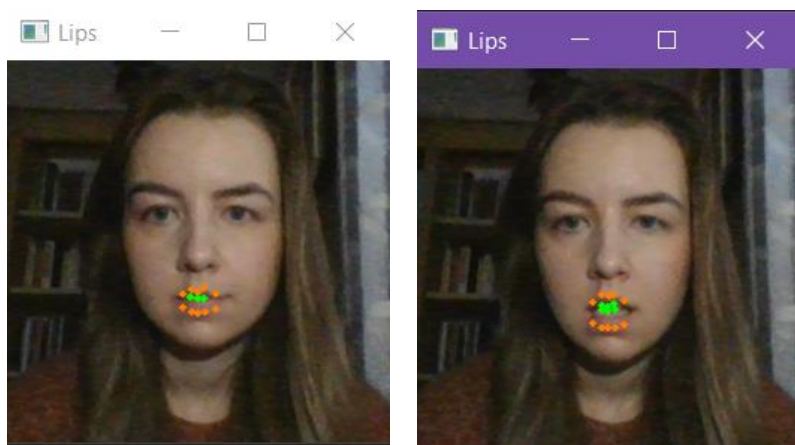
детекције које не треба санкционисати. И овај детектор садржи бафер који представља прозор фрејмова и функционише на сличан начин као претходно описани детектори. Оба ока су праћена истовремено и мерено је растојање зенице од централног положаја по хоризонтали (чиме се детектује гледање у лево или десно) и по вертикали (чиме се детектује гледање на горе). Поглед на доле није детектован на овај начин јер су тада очи практично затворене и зеница се не може детектовати, али је искоришћена информација о EAR која је претходно израчуната. Детектору се прослеђују поравната слика и карактеристичне тачке једног и другог ока. Прати се центар зенице (уз помоћ функција из OpenCV библиотеке) и рачуна се однос између ширине (висине) ока и растојања између леве (горње) границе ока и центра зенице. Овај однос је опсегу од 0 до 1, а док особа гледа право вредност је 0,5. Ако је однос по хоризонтали мањи од 0.35 сматра се да особа гледа лево, а ако је већи од 0.65 десно. Ако је однос по вертикали мањи од 0.35 сматра се да особа гледа на горе, а ако је EAR мањи од граничног сматра се да гледа на доле. Карактеристичне тачке очију се користе како би се изоловали са слике само региони од интереса. Први корак је издвајање ока, затим тражење одговарајућег прага како би се издвојила зеница и након тога налажење положаја зенице који се користи за одређивање смера гледања.

Око се са слике издваја тако што се направи једна црна слика истих димензија као фрејм, затим слика која служи као маска (бела слика на којој се црним пикселима попуни полигон који представља око) и након тога се примени функција *bitwise_not* која као параметре има црну слику (улазна), фрејм (излазна) и маску која специфицира који део излазне слике ће бити модификован инвертовањем улазне слике. Добија се бела слика на којој је само око у боји. Након тога је исечен само део те слике на коме је око. Затим се проналази праг (позивом функције *threshold*) који ће најбоље издвојити дужицу ока. Покушава се са вредностима између 5 и 100, са кораком 5, издваја се дужица и упоређује се проценат који она заузима у оку са просечним процентом дужице у оку од 0,48. Она се издваја тако што се примени функција *bilateralFilter* над издвојеним оком како би се уклонио нежељени шум и задржале ивице, затим се примени операција ерозије са кернелом 3×3 и ако је вредност пиксела мања од прага који се тестира, поставља се на 0, а у супротном на 255, тако да ће бити враћено око које је беле боје, са дужицом црне боје. Проценат који заузима дужица се рачуна као проценат црних пиксела на слици ока. Након проналажења најбољег прага користи се функција *findContours* за проналажење контуре дужице и *moments* како би се нашле координате центра те контуре, односно зенице [46]. На слици 37 приказан је пример гледања са стране са означеним центрима зеница.



Слика 37: Детекција гледања са стране

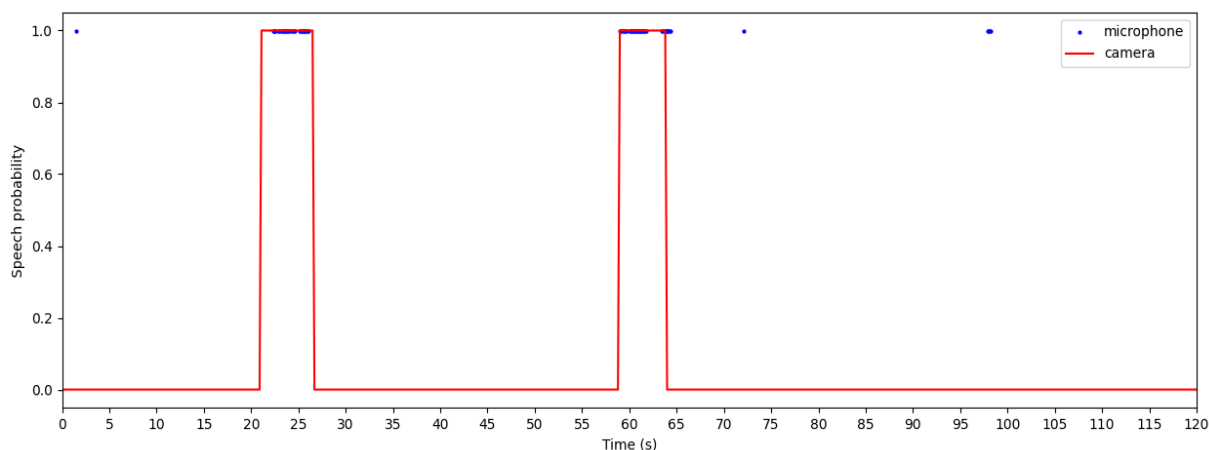
Детекција причања је реализована у оквиру модула *SpeechDetector* праћењем покрета усана. Коришћене су карактеристичне тачке горње и доње усне и мерена растојања између одговарајућих тачака. Детектор је иницијализован сликом ученика из базе података на којој су ученику уста затворена. Вредности које представљају растојања су унете у 2 низа: један за спољашње (означене наранџастом бојом), један за унутрашње (означене зеленом бојом) тачке приказане на слици 38. Ови низови су третирани као вектори, који су након нормализације упоређивани са истим низовима на фрејму. Ако је еуклидско растојање између њих веће од унапред задате вредности (експериментално је одабрана вредност 0.2) уста се сматрају отвореним. Креирани су бафери као код претходних детектора. У оквиру прозора фрејмова броје се фрејмови на којима су уста ученика отворена и уколико је при преласку на наредни прозор тај број већи од четвртине обрађених фрејмова пријављује се да је ученик причао. Како се може десити да се ученик прозева, смеје или из неког разлога држи уста отвореним дуже време, те ситуације се толеришу и не сматрају невалидним. Постоји бафер који чува узастопне фрејмове у којима су уста отворена. Уколико на преласку између прозора овај бафер има елемената, отварање новог прозора се одлаже док су уста отворена и ти фрејмови се сматрају валидним. Експериментално је утврђено да приликом причања неће бити више од 4 узастопна фрејма када су уста отворена, тако да се низови дужи од 4 сматрају валидним „зевањем“ и уколико се налазе у оквиру прозора неће се урачунавати у бројање. При ресетовању детектора се испитује да ли је обрађено више од две трећине текућег прозора и спорном ситуацијом се сматра случај када је бројач фрејмова са отвореним устима већи од половине броја обрађених фрејмова.



Слика 38: Спољашње (наранџасто) и унутрашње (зелено) тачке усана

Детекција би била поузданија ако би се користили подаци са микрофона у оквиру модула који би обрађивао аудио податке и детектовао интервале у којима има и гласовне активности. У овом пројекту је фокус био на праћењу података са камере, али је, поређења ради, покренута још једна нит у оквиру које је симултано детектован људски глас. Дакле, једна нит прихвата и обрађује податке са камере, а друга са микрофона. Уз помоћ *pyaudio* библиотеке отворен је аудио стрим и током трајања теста учитавање су секвенце аудио података у трајању од 20 ms које су прослеђиване детектору гласовне активности из *webrtcvad* библиотеке [47]. Ове секвенце се називају фрејмовима и најчешће су у трајању од 20 ms зато што је то оптималан интервал у оквиру кога се звучни сигнал може сматрати стационарним процесом над којим се могу примењивати алгоритми процесирања. На слици 39 је приказан график вероватноће присутности људског гласа у времену. Посматран је период од 120 секунди. Вероватноћа има

вредност 0 ако се сматра да није детектован глас, а 1 у супротном. Црвеном линијом је означен резултат добијен на основу обраде фрејмова са камере, док је плавим тачкама означен резултат добијен обрадом аудио фрејмова. Једна плава тачка представља вероватноћу за постојање гласовне активности у десетом делу секунде. Ова вредност је 1 уколико је више од 50% фрејмова из тог дела секунде детектор гласовне активности означио позитивно. Може се приметити да се добијени резултати у великој мери поклапају. Изоловане плаве тачке су ситуације у којима је глас трајао веома кратко или које су грешком означене као позитивне и које не треба посматрати као спорне. За цртање графика искоришћена је *matplotlib* библиотека.



Слика 39: График зависности вероватноће присутности гласовне активности од времена

На крају, у сврху препознавања лица у оквиру модула *FaceRecognizer* коришћен је истрениран модел резидуалне дубоке неуронске мреже из Dlib библиотеке, на сличан начин као у библиотеци *face_recognition* [48]. Овај модел има тачност од 99.38% на *Labeled Faces in the Wild* скупу слика. Коришћена је ResNet мрежа која генерише 128 јединствених реалних вредности за сваку улазну слику. Као прозор фрејмова коришћен је бафер величине 150 и у оквиру сваког прозора је вршено препознавање по три пута (са једнаким размацима) с обзиром на то да је ово захтевнија операција и проузроковала би знатно слабије перформансе када би била извршавана над сваким фрејмом. Коришћењем функције *compute_face_descriptor*, којој се као параметри дају поравната слика лица и карактеристичне тачке лица, израчунат је дескриптор за слику из базе. Лице са камере упоређивано је са овом сликом, тачније, упоређивани су њихови дескриптори и уколико је разлика већа од 0.5, сматра се да лице није препознато. Уколико су у прозору била бар два неуспешна препознавања тај прозор се означава као споран. Различита лица углавном имају разлику од преко 0.8, а врло слична лица између 0.6 и 0.7, те је зато као граница одабрана вредност 0.5. Разлика је рачуната као еуклидско растојање одговарајућих дескриптора (вектора). Нема потребе упоређивати улазну слику са свим сликама из базе. При ресетовању овог детектора, уколико је до момента ресета бар два пута у текућем прозору препознавање било неуспешно, фрејмови из тог прозора који су обрађени означавају се као нерегуларни.

5. Закључак

Развој интернет технологија омогућио је да онлајн учење буде све заступљеније у пракси. Платформе за учење посебно су добиле на значају у периоду пандемије која је спречила традиционално одржавање наставе у учионицама. Предности оваквог начина учења су бројне, почевши од могућности рада од куће до веома брзе и прегледне анализе резултата ученика. Са друге стране, многи сматрају да знање стечено кроз „уживо“ наставу никада неће моћи да буде једнако добро као знање добијено онлајн учењем.

Софтвер за полагање тестова је развијан заједно са софтверима за учење с обзиром на то да су тестови најчешћи начин вредновања знања. Надгледање може вршити дежурно особље, али вештачка интелигенција и машинско учење значајно су унапредили овај процес јер софтвери за аутоматско надгледање полагања у доброј мери омогућавају детекцију нерегуларности. Приликом тестирања могу се надгледати микрофон, камера и екран особе која полаже тест и најпрецизнија анализа добија се комбинацијом сва три извора информација. Многи софтвери омогућавају полагање тестова и на уређајима попут мобилних телефона, али надгледање најбоље функционише на десктоп или лаптоп рачунарима, тако да може бити проблема уколико ученици нису у могућности да их обезбеде.

У овом раду фокус је био на обради података са камере, дат је преглед техника рачунарског вида које се могу приметити у ту сврху и имплементиран је прототип система који примењује неке од њих. Алгоритми као што су Viola-Jones, HOG и Eigenfaces имали су велики значај за развој области детекције објеката и детекције лица. Ови алгоритми се извршавају веома брзо, али се не понашају довољно добро у условима слабијег осветљења и оријентације објеката (лица) која није фронтална. Ове недостатке превазилазе модернији методи засновани на дубоким неуронским мрежама, који су данас и најзаступљенији. За детекцију објеката најчешће се користе детектори као што су YOLO и SSD. Детекција и препознавање лица такође дају најбоље резултате када се користе неуронске мреже. За праћење покрета главе и очију најбоље би било користити 3D моделе, али задовољавајући резултати се могу добити и уз одређене апроксимације. Детекција разговора може се вршити праћењем покрета усана на снимку са камере, у комбинацији са детекцијом гласовне активности путем микрофона. Прототип система који је реализован је доста једноставнији од система који би могао да се користи у реалности. При одабиру алгоритама узето је у обзир окружење у коме систем треба да се извршава. Камера има увид у ограничен део простора, тако да поред особе која ради тест може седети неко и помагати јој, а да то не буде примећено. Овај случај би могао бити превазиђен уколико би се захтевао и снимак комплетне просторије у којој се полаже тест коришћењем додатне камере која снима у 360°. Међутим, скенирање собе ученика може се сматрати нарушавањем њихове приватности, због чега је било случајева протеста ученика. Провера да ли је лице „живо“ је вршена на основу бројања трептаја, што може довести до погрешних закључака уколико су неке очи природно затвореније него просечној особи, тако да би било боље одабрати неку од напреднијих метода. Такође, ако особа дужи период гледа на доле, то ће бити детектовано као спорна ситуација, али ће највероватније бити и лажно детектовано да у том периоду лице није било „живо“ јер неће бити довољно детектованих трептаја.

Системи за аутоматско надгледање полагања тестова дају прилично добре резултате, али увек се могу наћи начини да се систем превари. Такође, алгоритми који се примењују раде са одређеном грешком тако да се још увек не можемо у потпуности ослонити на њих и искључити људски фактор, али чињеница је да су ови системи све напреднији и применљивији.

Литература

- [1] S. Ryan, „Why Live Proctoring of Online Exams Is Becoming Mainstream,“ јун 2022. [Онлајн]. Доступно: https://www.testreach.com/blog-post/proctoring-online-exams.html?fbclid=IwAR0r8C4Cg4BLoVv-VJdbadDFxZk8qAiA_HHEkq7mLrEGkpa-U_7iah8jG10. [Приступано: 01.10.2022].
- [2] A. Sharma, „12 Best Proctoring Software for Cheating-Free Online Exams,“ август 2022. [Онлајн]. Доступно: <https://www.techjockey.com/blog/proctoring-software-for-online-exam>. [Приступано: 01.10.2022].
- [3] G. Brown, „Can An Online Exam Detect Cheating?,“ 2022. [Онлајн]. Доступно: https://www.masterteachingonline.com/can-an-online-exam-detect-cheating/?fbclid=IwAR0CTcD3-2ixdQQaBXkrii_eVHSyffVz2AZ4P08q0tc4cSRFCm0-VVLCOPs. [Приступано: 01.10.2022]
- [4] Honorlock, „Exclusive Remote Proctoring Features“. [Онлајн]. Доступно: <https://honorlock.com/exclusive/>. [Приступано: 06.10.2022].
- [5] Mercer | Mettl, „Proctored Exams: Secure Online Proctoring With AI-Powered Tools“. [Онлајн]. Доступно: <https://mettl.com/online-remote-proctoring>. [Приступано: 06.10.2022].
- [6] „Computer vision,“ у *Wikipedia, the Free Encyclopedia*. Доступно: https://en.wikipedia.org/wiki/Computer_vision. [Приступано: 23.10.2022].
- [7] „Object detection,“ у *Wikipedia, the Free Encyclopedia*. Доступно: https://en.wikipedia.org/wiki/Object_detection. [Приступано: 24.10.2022].
- [8] G. Boesch, „Object Detection in 2022: The Definitive Guide,“ 2022. [Онлајн]. Доступно: <https://viso.ai/deep-learning/object-detection/>. [Приступано: 24.10.2022].
- [9] J. Hui, „mAP (mean Average Precision) for Object Detection,“ март 2018. [Онлајн]. Доступно: <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>. [Приступано: 04.12.2022].
- [10] P. Viola, M. Jones, „Rapid Object Detection using a Boosted Cascade of Simple Features,“ у *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001. Доступно: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>. [Приступано: 26.10.2022].
- [11] N. Dalal, B. Triggs, „Histograms of Oriented Gradients for Human Detection,“ у *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005. Доступно: <http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf>. [Приступано: 28.10.2022].
- [12] S. Mallick, „Histogram of Oriented Gradients explained using OpenCV,“ децембар 2016. [Онлајн]. Доступно: <https://learnopencv.com/histogram-of-oriented-gradients/>. [Приступано: 31.10.2022].
- [13] R. Gandhi, „Support Vector Machine - Introduction to Machine Learning Algorithms,“ јун 2018. [Онлајн]. Доступно: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>. [Приступано: 31.10.2022].
- [14] R. Girshick, J. Donahue, T. Darrell, J. Malik, „Rich feature hierarchies for accurate object detection and semantic segmentation,“ у *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014. Доступно: <https://arxiv.org/pdf/1311.2524.pdf>. [Приступано: 01.11.2022].

- [15] М. Дабовић, И. Тартаља, „Дубоке конволуцијске неуронске мреже – концепти и актуелна истраживања“, у *Зборник 61. Конференција за електронику, телекомуникације, рачунарство, аутокатику и нуклеарну технику, ЕТРАН 2017*, 2017. Доступно: https://www.researchgate.net/publication/321709294_Duboke_konvolucijske_neurons_ke_mreze_-_koncepti_i_aktuelna_istrazivanja. [Приступано: 01.11.2022].
- [16] R. Girshick, „Fast R-CNN“, у *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. Доступно: <https://arxiv.org/pdf/1504.08083.pdf>. [Приступано: 04.11.2022].
- [17] S. Ren, K. He, R. Girshick, J. Sun, „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks“, у *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, 2015. Доступно: <https://arxiv.org/pdf/1506.01497.pdf>. [Приступано: 04.11.2022].
- [18] K. He, G. Gkioxaris, P. Dollár, R. Girshick, „Mask R-CNN“, у *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. Доступно: <https://arxiv.org/pdf/1703.06870.pdf>. [Приступано: 04.11.2022].
- [19] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, „You Only Look Once: Unified, Real-Time Object Detection“, у *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Доступно: <https://arxiv.org/pdf/1506.02640.pdf>. [Приступано: 09.11.2022].
- [20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. C. Berg, „SSD: Single Shot MultiBox Detector“, у *European Conference on Computer Vision (ECCV) (2016)*, 2016. Доступно: <https://arxiv.org/pdf/1512.02325.pdf>. [Приступано: 10.11.2022].
- [21] J. Hui, „SSD object detection: Single Shot MultiBox Detector for real-time processing“, март 2018. [Онлајн]. Доступно: <https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06>. [Приступано: 10.11.2022].
- [22] „Face detection“, у *Wikipedia, the Free Encyclopedia*. Доступно: https://en.wikipedia.org/wiki/Face_detection. [Приступано: 10.11.2022].
- [23] „Facial recognition system“, у *Wikipedia, the Free Encyclopedia*. Доступно: https://en.wikipedia.org/wiki/Facial_recognition_system. [Приступано: 10.11.2022].
- [24] A. Rosebrock, „Liveness Detection with OpenCV“, март 2019. [Онлајн]. Доступно: <https://pyimagesearch.com/2019/03/11/liveness-detection-with-opencv/>. [Приступано: 09.11.2022].
- [25] A. Rosebrock, „What is face recognition?“, мај 2021. [Онлајн]. Доступно: <https://pyimagesearch.com/2021/05/01/what-is-face-recognition/>. [Приступано: 09.11.2022].
- [26] M. A. Turk, A. P. Pentland, „Face Recognition Using Eigenfaces“, у *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991. Доступно: <https://sites.cs.ucsb.edu/~mturk/Papers/mturk-CVPR91.pdf>. [Приступано: 10.11.2022].
- [27] „ML | Face Recognition Using Eigenfaces (PCA Algorithm)“, септембар 2021. [Онлајн]. Доступно: <https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/>. [Приступано: 10.11.2022].
- [28] P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman, „Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection“, у *IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 19, Issue: 7, July 1997)*, 1997. Доступно: <https://web.ece.ucsb.edu/~hespanha/published/faces.pdf>. [Приступано: 10.11.2022].

- [29] T. Ahonen, A. Hadid, M. Pietikäinen, „Face Recognition with Local Binary Patterns,“ у *Computer Vision - ECCV 2004, 8th European Conference on Computer*, 2004. Доступно: https://link.springer.com/content/pdf/10.1007/978-3-540-24670-1_36.pdf. [Приступано: 10.11.2022].
- [30] F. Schroff, D. Kalenichenko, J. Philbin, „FaceNet: A Unified Embedding for Face Recognition and Clustering,“ у *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Доступно: https://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf. [Приступано: 01.12.2022].
- [31] A. Geitgey, „Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning,“ јул 2016. [Онлајн]. Доступно: <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffe121d78>. [Приступано: 10.11.2022].
- [32] S. Mallick, „Head Pose Estimation using OpenCV and Dlib,“ септембар 2016. [Онлајн]. Доступно: <https://learnopencv.com/head-pose-estimation-using-opencv-and-dlib>. [Приступано: 11.11.2022].
- [33] S. Mallick, „Gaze Tracking and Estimation,“ *YouTube*, новембар 2019. [Видео датотека]. Доступно: <https://www.youtube.com/watch?v=-lmc2-podgQ>. [Приступано: 11.11.2022].
- [34] M. Fabien, „Voice Activity Detection“. [Онлајн]. Доступно: https://maelfabien.github.io/machinelearning/Speech4/?fbclid=IwAR2V3yKH5REzQLgUDfCkN1X52fkQ10io7H_4Q-iKCrtOcJyhuPwXlB4wwQ#. [Приступано: 12.11.2022].
- [35] „Speech recognition,“ у *Wikipedia, the Free Encyclopedia*. Доступно: https://en.wikipedia.org/wiki/Speech_recognition. [Приступано: 12.11.2022].
- [36] L. Xia, G. Chen, X. Xu, J. Cui, Y. Gao, „Audiovisual speech recognition: A review and forecast,“ децембар 2020. [Онлајн]. Доступно: <https://journals.sagepub.com/doi/full/10.1177/1729881420976082#fig3-1729881420976082>. [Приступано: 04.12.2022].
- [37] OpenCV, „TensorFlow Object Detection API“. [Онлајн]. Доступно: <https://github.com/opencv/opencv/wiki/TensorFlow-Object-Detection-API>. [Приступано: 28.10.2022].
- [38] A. Rosebrock, „Object detection with deep learning and OpenCV,“ септембар 2017. [Онлајн]. Доступно: <https://pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/>. [Приступано: 20.11.2022].
- [39] T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, P. Dollár, „Microsoft COCO: Common Objects in Context,“ у *European Conference on Computer Vision*, 2014. Доступно: <https://arxiv.org/pdf/1405.0312.pdf>. [Приступано 28.10.2022].
- [40] A. Rosebrock, „Face detection tips, suggestions, and best practices,“ април 2021. [Онлајн]. Доступно: <https://pyimagesearch.com/2021/04/26/face-detection-tips-suggestions-and-best-practices/>. [Приступано: 04.10.2022].
- [41] [Онлајн]. Доступно: https://github.com/opencv/opencv/tree/4.x/samples/dnn/face_detector. [Приступано: 04.10.2022].
- [42] [Онлајн]. Доступно: http://dlib.net/face_landmark_detection.py.html. [Приступано: 01.10.2022].
- [43] C. Sagonas, S. Zafeiriou, „Facial point annotations“. [Слика]. Доступно: <https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/>. [Приступано: 01.10.2022].

- [44] A. Rosebrock, „Face Alignment with OpenCV and Python,“ мај 2017. [Онлајн]. Доступно: <https://pyimagesearch.com/2017/05/22/face-alignment-with-opencv-and-python/>. [Приступано: 13.11.2022].
- [45] T. Soukupová, J. Čech, „Real-Time Eye Blink Detection using Facial Landmarks,“ у *21st Computer Vision Winter Workshop*, 2016. Доступно: <http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf>. [Приступано: 02.11.2022].
- [46] A. Lamé, „Gaze Tracking,“ 2019. [Онлајн]. Доступно: <https://github.com/antoinelame/GazeTracking>. [Приступано: 14.11.2022].
- [47] J. Wiseman, „py-webrtcvad,“ 2016. [Онлајн]. Доступно: <https://github.com/wiseman/py-webrtcvad>. [Приступано: 24.11.2022].
- [48] A. Geitgey, „Face Recognition,“ 2021. [Онлајн]. Доступно: https://github.com/ageitgey/face_recognition#face-recognition. [Приступано: 14.11.2022].