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.

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

Комисија:

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Датум пријаве: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Датум предаје: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Датум одбране: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ниш, 2022.

**Садржај**

[**Увод** 3](#_Toc118050246)

[**Софтвери за праћење регуларности полагања тестова** 4](#_Toc118050247)

[Аутоматско надгледање онлајн тестирања 4](#_Toc118050248)

[Примери софтвера за аутоматско надгледање онлајн тестирања 5](#_Toc118050249)

[**Технике рачунарског вида за праћење регуларности полагања тестова** 7](#_Toc118050250)

[Детекција објеката 7](#_Toc118050251)

[**Viola-Jones (Haar) детектор** 8](#_Toc118050252)

[**HOG детектор** 10](#_Toc118050253)

[**R-CNN детектор** 13](#_Toc118050254)

[**YOLO детектор** 16](#_Toc118050255)

[**SSD детектор** 17](#_Toc118050256)

[Препознавање лица 20](#_Toc118050257)

[**Eigenfaces** 21](#_Toc118050258)

[**Fisherfaces** 22](#_Toc118050259)

[**LBP** 23](#_Toc118050260)

[**Дубоко учење** 24](#_Toc118050261)

[Одређивање положаја главе 24](#_Toc118050262)

[Праћење покрета очију 26](#_Toc118050263)

[Детекција причања 26](#_Toc118050264)

# **Увод**

Пандемија настала услед ширења вируса Ковид-19 утицала је на све аспекте живота људи, укључујући образовање. У условима који су настали као последица ове ситуације било је неопходно прилагодити начин извођења наставе како би се школска година одвијала несметано. Многе образовне институције биле су приморане да пређу на хибридни или у потпуности онлајн модел одржавања наставе, те је један од највећих изазова био како пронаћи најбољи начин за оцењивање ученика.

Онлајн тестирање је већ било у великој мери заступљено у пракси зато што олакшава и убрзава процес креирања тестова, полагања, као и анализу резултата. Флексибилност и могућност рада од куће разлози су због којих је велики број наставника одабрао овакве тестове као меру оцене. Наравно, тиме се отвара простор за различите начине преписивања. Неки од начина да се преписивање спречи 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 и ученици ипак нађу начин да комуницирају са неким или пронађу одговоре у литератури или на интернету. Из тог разлога направљени су софтвери за надгледање полагања онлајн тестирања. У наставку ће се учеником ословљавати особа која полаже тест, али ови софтвери, поред примене у образовним институцијама, имају примену и у различитим компанијама и организацијама за потребе интервјуа за посао, семинара, добијања сертификата...

Најчешће се прате микрофон, камера и екран ученика. Једно од решења је да дежурна особа надгледа ученика током полагања у реалном времену. Друго решење подразумева снимање ученика током полагања и касније прегледање тог снимка ради контроле регуларности. Предност оваквог решења је што ученик и дежурни не морају у заказано време бити присутни, али у овом случају нема могућности да дежурна особа реагује у реалном времену на неправилности због којих је можда требало онемогућити наставак полагања теста ученику. Треће решење је аутоматско нагледање полагања. Коришћење алгоритама вештачке интелигенције и машинског учења у великој мери мења особу која би била задужена да лично надгледа снимке ученика који полажу тестове. Систем генерише упозорења током полагања и бележи све нерегуларне ситуације које су детектоване и које се касније могу прегледати како би се потврдиле неправилности. [1]

У овом раду биће обрађене методе које се могу користити за обраду података са камере приликом полагања онлајн тестова. Рад је организован у пет поглавља. Након уводног дела, у другом поглављу је дат преглед постојећих софтверских решења за проблем надгледања онлајн тестова. У трећем поглављу прво су обрађене технике и алгоритми рачунарског вида који се могу применити за детекцију објеката, на које се надовезују алгоритми за детекцију и препознавање лица. Затим су описани начини за одређивање положаја главе, праћење покрета очију и детекцију причања на видео снимку. У наредном поглављу описана је конкретна имплементација једноставног прототипа система који се може користити за надгледање регуларности полагања онлајн тестова на основу фрејмова са камере. Систем детектује нерегуларне ситуације (у виду одсуства особе у кадру или присуства више особа, коришћења мобилног телефона, окретања и гледања са стране, разговора), врши препознавање лица, приказује упозорења ученику за време теста и све детектоване нерегуларности бележи у видео фајл. Закључно поглавље осврће се на претходна поглавља и сумира теоријске и практичне аспекте описаног проблема.

# **Софтвери за праћење регуларности полагања тестова**

Постоје различити софтвери који су развијени у циљу омогућавања реализације учења на даљину. Неки од њих укључују и могућност тестирања и оцењивања, али постоје и посебне платформе само за ову намену. Углавном се ради о веб апликацијама, а има и апликација које је потребно инсталирати на свом рачунару или мобилном телефону. За креирање кратких тестова који не захтевају висок степен сигурности и провере, најједноставније је користити неки од бесплатних веб сајтова за генерисање квизова и тестова који немају никакав вид контроле. Када су у питању озбиљнији тестови који захтевају висок степен сигурности, треба одабрати неки од напреднијих софтвера који нуде надгледање регуларности полагања, у зависности од потреба и буџета организације.

Као најважнији захтеви које је потребно да напреднији софтвери за онлајн тестирање испуне могу се издвојити: једноставност коришћења, превенција варања, сигурност, скалабилност, прилагодљивост и интеграција са другим сервисима. Једноставност коришћења је веома важна јер софтвер треба да корисницима пружи добро корисничко искуство. Технике за детекцију нерегуларности су неопходне како би се очувао интергритет теста и како би тестирање било еквивалентно тестирању у заједничкој просторији у којој једна или више особа дежура. Потребно је обезбедити сигурност осетљивих података као што су сами тестови и лични подаци корисника. Скалабилност има важну улогу када је у питању употребљивост система зато што може бити потребно да више хиљада корисника истовремено полаже тестове. Како корисници апликације могу бити различите институције и организације, имаће различите потребе, те треба обезбедити могућност избора функционалности које су захтеване за реализацију одређеног тестирања. Још једна важна особина је да систем треба да буде једноставан за интеграцију са постојећим платформама за учење. [2]

Када је реч о софтверима за надгледање онлајн тестирања, у уводном поглављу је напоменуто да се они могу поделити у три групе. У прву групу спадају софтвери код којих особа која је задужена за надгледање регуларности уживо посматра ученика који полаже тест и може одмах реаговати уколико примети неке неправилности. Систем захтева од ученика да укључи камеру, микрофон и подели свој екран како би дежурна особа имала увид у све његове активности. Други начин за реализацију система је да се током полагања аудио, видео и учеников екран снимају, а касније прегледа убрзана верзија тог снимка. Софтвери из треће групе не захтевају ангажовање дежурне особе током тестирања, користе се у реалном времену, засновани су на техникама вештачке интелигенције и све више се користе у пракси. У наставку ће бити дат преглед начина функционисања софтвера за аутоматско надгледање онлајн тестирања и неколико популарних производа који су доступни и користе их многобројне институције и организације.

## **Аутоматско надгледање онлајн тестирања**

Први корак у процесу полагања тестова углавном је регистровање корисника које може бити имплементирано на различите начине, али најчешће се ученик пријављује на систем коришћењем свог и-мејла (енг. email), а пре самог покретања теста може се од њега затражити да коришћењем камере покаже своју идентификациону картицу или омогући систему да забележи његову слику и упореди је са сликом из базе података. У оквиру апликације може постојати више модула који су задужени за обраду различитих података. 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 и да ли се поклапа са гласом тог ученика из базе података, као и да ли ученик разговара са неким. Надгледање звука може спречити да неко други ради тест уместо ученика и да му нека особа или апликација помажу „добацивајући“ му одговоре на питања. Модул који је задужен за праћење екрана ученика углавном санкционише сликање и снимање прозора у ком је отворен тест, напуштање теста и отварање других страница, апликација или фајлова. Првенствено је циљ забранити отварање апликација које могу олакшати дељење одговора. Модул за обраду података са камере захтева да камера ученика буде укључена током полагања теста и може да контролише да ли у кадру има других особа, недозвољених уређаја попут мобилних телефона, литературе, да врши препознавање лица, могу се пратити покрети ученика и детектовати када он гледа са стране. Софтвер за надгледање полагања додатно може вршити и онлајн претрагу како би проверио да ли је садржај теста „процурео“. Поједини софтвери су у могућности да уз помоћ вештачке интелигенције лоцирају и уклоне садржај теста који је неауторизовано доспео на интернет. [3]

## **Примери софтвера за аутоматско надгледање онлајн тестирања**

Једно од водећих решења је Honorlock, први сервис за надгледање онлајн тестирања који комбинује аутоматско надгледање са људским дежурањем. Тестирање надгледа вештачка интелигенција и ако детектује академску нечеститост алармира дежурну особу да се прикључи сесији у реалном времену. Овиме се губи потреба да нека особа све време надгледа више кандидата, што и кандидатима смањује осећај нелагодности због присуства друге особе. Предуслови за израду теста су да особа која полаже тест буде сама у просторији, да буде тишина, да рачунар на коме се ради тест има само један монитор и да корисник поседује 360° камеру како би се скенирала просторија у којој се налази. Honorlock користи софтвер за закључавање веб претраживача који онемогућава приступ другим веб сајтовима, при чему закључава и одређене пречице на тaстатури, онемогућава минимизирање претраживача и напуштање теста пре предаје. Наставници имају могућност да по жељи специфицирају којим веб сајтовима желе да омогуће приступ ученицима током тестирања. Могућа је директна интеграција са платформама за учење. Такође, софтвер идентификује „процурели“ садржај теста на интернету и предузима кораке за његово уклањање. Још једна функционалност која је уведена је да систем детектује ако ученик покуша да приступи материјалу за учење током теста путем неког другог уређаја и бележи снимак екрана током трајања приступа. Honorlock не користи биометријске методе за идентификацију ученика попут препознавања лица, већ пре почетка теста услика ученика који држи своју идентификациону картицу и након 60 секунди омогућава полагање теста. [4]

ProctorEdu је још једно веб решење за онлајн надгледање, снимање и евалуацију корисничког понашања током онлајн тестирања, без потребе преузимања било каквог софтвера. Функционалност надгледања је интегрисана са платформама за учење и омогућава надгледање удаљеног тестирања уживо или у аутоматском моду. Софтвер нуди биометријску аутентификацију, подршку за Android и iOS мобилне уређаје, аутоматски опоравак након губљења конекције, могућност повезивања додатне камере са мобилног телефона за преглед простора где се налази ученик у 360° (камера се повезује скенирањем QR кода). Систем дежурне особе обавештава о нерегуларним активностима у реалном времену и омогућава да ученици комуницирају са њима путем чета, видеа или аудија уколико буду имали неких питања. Систем нуди функционалност препознавања лица током трајања теста, детекцију буке, контролише покушај претраге на интернету. Извештај о нерегуларностима се генерише у видео и пдф формату. [5]

Mercer **|** Mettl такође нуди безбедно тестирање. Овај софтвер користи више од 150 универзитета широм света и обављено је више од 12 милиона тестова ове године. Нуди избор између надгледања уживо и аутоматског надгледања. Може се користити као комплетна платформа за тестирање, а и као сервис за удаљено надгледање тестирања којим се проширује нека друга платформа за тестирање. Систем врши видео и аудио надгледање. Детектује уколико кандидат није присутан или је присутна особа која није она којом се представља, присуство мобилних телефона и других особа и разговор са другим особама. Студент се пре почетка теста пријављује на систем, а током трајања теста се у одређеним временским интервалима скенира слика студента и упоређује са његовом сликом из базе података. Тестови се могу полагати и на уређајима попут мобилног телефона и таблета, али за тестове који захтевају већу сигурност препоручљиво је полагати их на лаптоп или десктоп рачунарима. У понуди је и Mettl Secure Browser који искључује све екстерне портове, спречавајући кандидата да повеже секундарни екран. Browser такође искључује све софтвере за дељење садржаја и вебсајтове и не допушта напуштање прозора у коме се ради тест пре предаје теста. AI модул је обучен да детектује до 18 типова нерегуларности. [6]

Examus је један од најфлексибилнијих сервиса за онлајн надгледање. Користи веб камеру за препознавање лица и детекцију емоција, праћење погледа, детекцију других особа. Може се користити на мобилним телефонима. Нуди два главна производа: Proctor AI – софтвер за онлајн надгледање коришћењем вештачке интелигенције и Examus EQ – алат који надгледа пажњу и ангажовање особа током похађања онлајн курсева. Proctor AI комбинује податке из три извора (видео, аудио и радна површина рачунара) и онда их анализира коришћењем алгоритама базираних на неуронским мрежама како би креирао детаљан извештај о корисниковом понашању. Образовне институције, као клијенти, добијају тај извештај праћен видео доказом. У извештају се налазе линкови који нас одводе директно до релевантног дела видео доказа где можемо лично проверити шта се дешавало у том моменту. Бележе се ситуације попут гледања ван екрана, причања, комуницирања са другим особама, промена прозора на екрану, покушајa дељења екрана или звука. Постоји могућност укључивања дежурне особе која ће у реалном времену проверити забележене нерегуларности. [7]

# **Технике рачунарског вида за праћење регуларности полагања тестова**

Рачунарски вид је поље рачунарске науке које ради на томе да омогући рачунарима да виде, идентификују и обрађују дигиталне слике на сличан начин као што то чини људски вид. Тежи се разумевању и аутоматизацији задатака које визуелни систем човека може да уради. Као научна дисциплина, рачунарски вид се бави теоријом вештачких система која издваја информације из слика. Као технолошка дисциплина, настоји да примени теорије и моделе за изградњу система рачунарског вида. [8]

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 и праћење покрета очију и усана особе која полаже тест.

## **Детекција објеката**

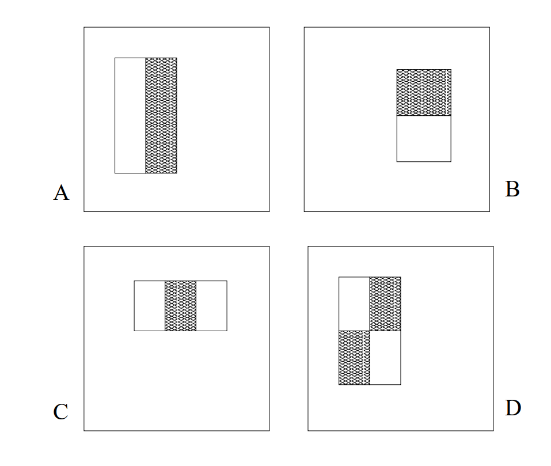
Детекција објеката је рачунарска технологија која спада у област рачунарског вида и обраде слика и бави се проналажењем објеката који припадају одређеним класама (нпр. људи, зграде, аутомобили) на дигиталним сликама и видео снимцима. [9] Свака класа објеката има специфичне особине, такозване фичере (енг. features), који одређују припадност објекта тој класи. Детекција објеката је један од основних проблема у рачунарском виду и има широку примену у свакодневном животу (нпр. за детекцију пешака и аутомобила, препознавање регистарских таблица, анализу слика и снимака у спорту и медицини). Даје нам информацију о томе који објекат је на слици и где се он налази.

Методе за детекцију објеката се генерално 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 неопходно је издвојити фичере објеката коришћењем неког од детектора фичера и након тога користити неку од техника за класификацију, попут SVM (енг. Support Vector Machine). Најпознатији примери ових метода су Viola–Jones детектор и хистограм оријентисанх градијената (HOG, енг. Histogram of Оriented Gradients). Технике базиране на дубоком учењу најчешће се ослањају на конволуционе неуронске мреже (CNN, енг. Convolutional Neural Networks). Детекција објеката захтева обављање два задатка: први је проналажење произвољног броја региона објеката, а други задатак је класификација сваког од њих и одређивање правоугаоника којим се објекат може уоквирити. Поступак се може реализовати кроз две фазе, а може и у оквиру једног корака како би се добиле боље перформансе. Детектори који раде у 2 фазе имају велику поузданост, али су спорији. Најпознатији двофазни детектори су R-CNN, Fast R-CNN, Faster R-CNN и Mask R-CNN. Детектори који раде у једном кораку предвиђају оквирне правоугаонике на слици без претходно издвојених региона од интереса третирајући детекцију објеката као регресиони проблем. Бржи су, структурно једноставнији и могу се користити код апликација од којих се очекује да раде у реалном времену, али су и нешто мање поуздани од једнофазних алгоритама. Најпознатији детектори из ове групе су YOLO (енг. You Only Look Once) који има више верзија и SSD (енг. Single Shot MultiBox Detector). [10]

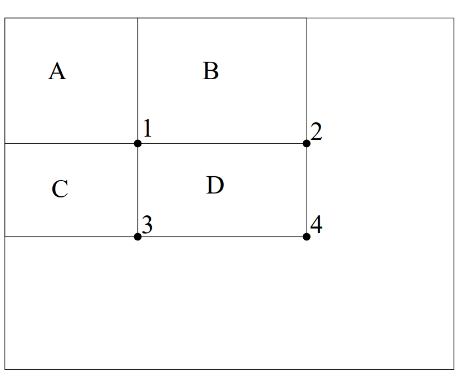
Раније методе за детекцију могу бити ограничене услед постојања комплексне позадине, делимично прекривених објеката, лошег осветљења и шума, док су технике дубоког учења су значајно отпорније на наведене проблеме. Данас су рачунари много ефикаснији него што је то било раније и тежи се паралелизацији процеса, тако да 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 великим бројем тренинг слика. У наставку ће бити дат преглед неколико детектора објеката који су били значајни за развој ове области, почевши од Viola-Jones детектора који се први појавио, преко HOG детектора који се појавио пар година касније, до детектора који користе неуронске мреже као што су различите верзије R-CNN-a, YOLO и SSD.

### **Viola-Jones (Haar) детектор**

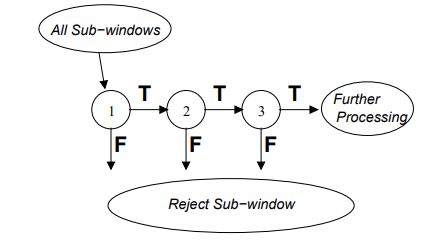
Paul Viola и Michael Jones су у свом раду [11] из 2001. године представили метод за детекцију објеката у реалном времену реализован кроз примену неколико нових техника. Првобитно је био намењен аутоматској детекцији лица, али може се користити за детекцију било ког типа објеката.

За детекцију је коришћен скуп фичера приказаних на слици 1 који подсећају на Haar-ове таласе[[1]](#footnote-1) (одакле и потиче назив детектора). Дефинисане су три врсте фичера: фичери са два (A и B) суседна правоугаоника исте величине која су хоризонтално или вертикално поравната, 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. Прва два фичера на слици (A и B) се користе за детекцију ивица, трећи (C) за детекцију линија, а четврти (D) за детекцију косих линија.

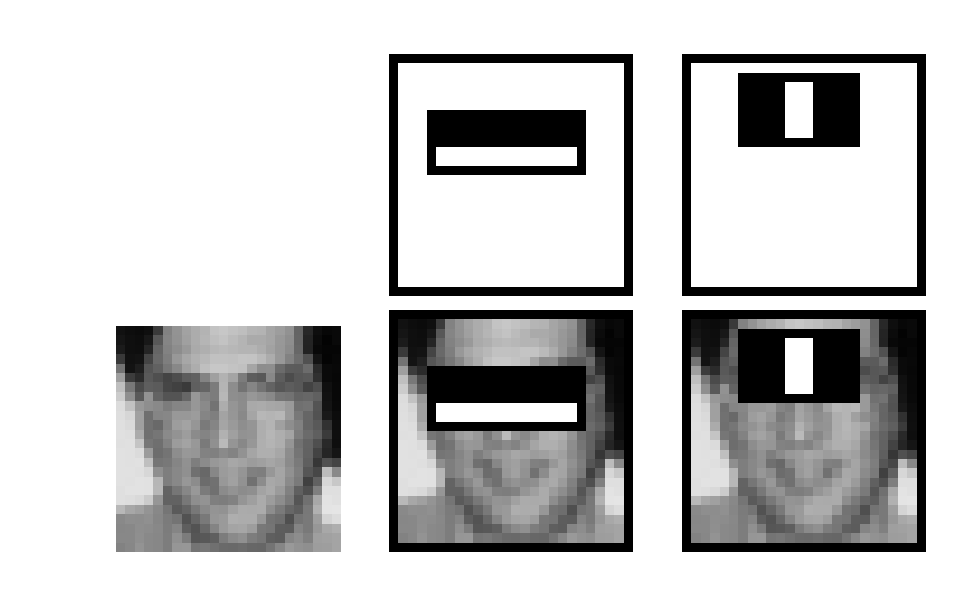
Слика 1: Haar фичери [11]

Аутори су приказали нови начин за репрезентацију слике назван „интегрална слика“, која омогућава да се фичери који се користе при детекцији брзо израчунају. Интегрална слика може бити израчуната 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 са оригиналне слике изнад и лево од тачке (x, y). Омогућава израчунавање суме пиксела произвољног правоугаоног региона коришћењем вредности у само 4 тачке. На слици 2 се може видети 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 А. Вредност у тачки 2 је , у тачки 3 је , а у тачки 4 је . Сума пиксела у оквиру правоугаоника D би била . За два правоугаоника сума може бити израчуната на основу вредности у 6 тачака, за случај 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 [11]

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 се издвојио мањи скуп најважнијих фичера из великог скупа фичера коришћен је алгоритам заснован на *AdaBoost* техници за учење. Свака фаза *boosting* процеса даје нови слаби класификатор који зависи само од једног фичера (бира се један од 180 000 фичера који има најмању грешку класификације). Овиме се број фичера смањује на око 6000. Слаби класификатори не могу самостално да класификују слику, али њихова линеарна комбинација може представљати 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), бива одбачен 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. Циљ је брзо одбацити регионе без објеката и даље процесирање вршити само над регионима који су обећавајући, односно који потенцијално садрже објекат.

Слика 3: Каскада класификатора [11]

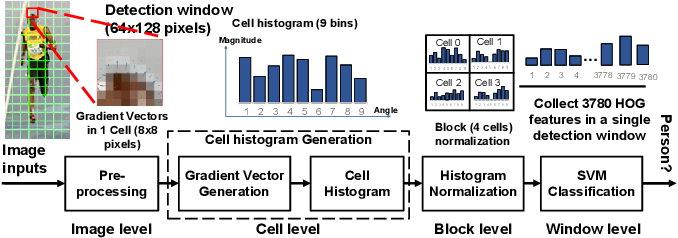
У оригиналном раду је детекција објеката демонстрирана на примеру детекције лица. На слици 4 приказана су прва два фичера селектована *AdaBoost* алгоритмом. Први фичер наглашава особину лица да је регион око очију углавном тамнији од региона горњег дела образа, док други фичер осликава особину да је регион очију тамнији од региона носа. Сваки од класификатора у каскади од 38 слојева трениран је коришћењем AdaBoost процедуре 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 потпрозори 24×24. У процесу детекције детектор обилази слику која се тестира методом клизајућег прозора у више величина, при чему се детектор (прозор) скалира, а не слика. 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 подскупове, 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. [11]

Слика 4: Прва два фичера селектована AdaBoost техником [11]

Овај алгоритам ради са сивим (енг. grayscale) сликама, детектује објекте на сликама независно од њихове локације и величине и инваријантан је на промене у осветљењу. Једна од највећих предности овог детектора у односу на остале је велика брзина детекције, а недостатак што је склон лажно позитивним детекцијама и детектује углавном само фронтално постављене објекте. Иако је објављен пре доста година, и даље се често користи када се детекција врши на ограниченим уређајима на којима не можемо у реалном времену извршавати алгоритме који су рачунски захтевнији.

### **HOG детектор**

У раду [13] из 2005. године показано је да се за детекцију објеката на слици могу успешно користити HOG фичер дескриптор слике и линеарни SVM класификатор. Аутори су се фокусирали на детекцију пешака на слици, али детектор се може применити и за детекцију објеката других класа.

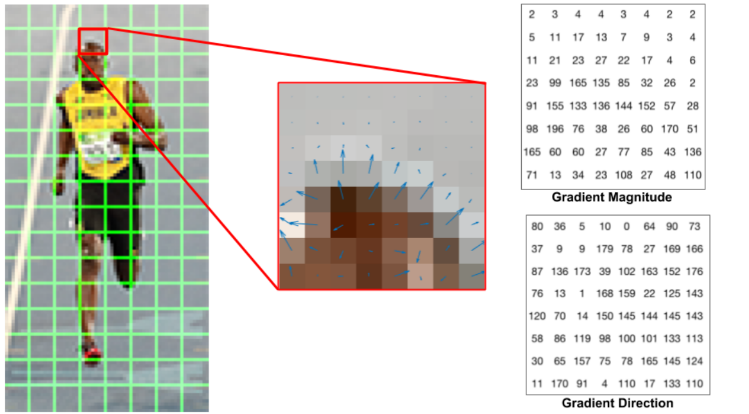
Фичер дескриптор је репрезентација слике која издваја корисне информације из ње, а одбацује ирелевантне. Код HOG фичер дескриптора се улазна слика димензија 64 × 128 × 3 (канала) конвертује у вектор фичера дужине 3780, а као фичери користе се хистограми смерова оријентисаних градијената. Градијенти слике (x и y изводи) су корисни зато што је магнитуда градијената велика око ивица и ћошкова (региони наглих промена интензитета), а они носе много више информација о облику објекта него равни региони. На слици 5 приказан је процес детекције објеката HOGдетектором*.* Први корак у рачунању хистограма оријентисаних градијената 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. Посматрају се региони слике са различитим скалирањем и на различитим локацијама. Једино ограничење је да парче које се обрађује има фиксни *aspect ratio* (у нашем случају 1:2). Дакле, са слике се издвоји део који има овај *aspect ratio* и скалира се на 64×128 пиксела.

Слика 5: Начин рада HOG детектора [14]

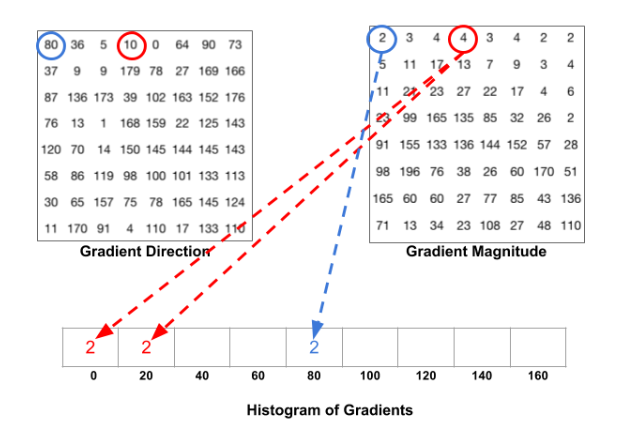
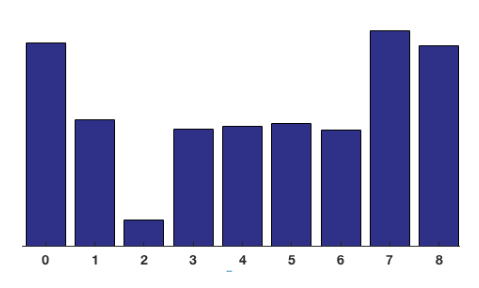
Затим је потребно израчунати хоризонталне и вертикалне градијенте, што се може постићи филтрирањем слике Собел филтером са кернелом величине 1. Магнитуда и угао се рачунају по следећим формулама: , , где су Gx и Gy градијенти по x и y осама. На слици 6 можемо приметити да x-градијент издваја вертикалне линије, а y-градијент хоризонталне. Магнитуда градијената се јавља где год постоје нагле промене у интензитету, тако да финална слика садржи истакнуте све ивице. Особа је у првом плану, а већина ирелевантних информација је уклоњена. У сваком пикселу градијент има магнитуду и смер. 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 6: x-извод слике (лево), y-извод слике (центар), магнитуда извода (десно) [14]

Затим се слика дели на ћелије од 8×8 пиксела и хистограм градијената се рачуна за сваку ћелију. Део слике 8×8 садржи вредности пиксела. Градијент овог дела садржи 2 вредности (магнитуду и смер) за сваки пиксел, што даје вредности. Индивидуални градијенти пиксела могу садржати шум, али хистограм над регионом од 8×8 пиксела је много отпорнији на шум. Ове димензије слике и ћелија су одабране за потребе детекције пешака и биле су довољно велике да се издвоје значајни фичери. На слици 7 приказана је подела слике на ћелије (лево). 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 су између 0° и 180° уместо 0° и 360° зато што су посматрани „неозначени“ градијенти код којих су градијент и његова негативна варијанта представљени истим бројем.



Слика 7: 8×8 ћелије (лево), део слике са градијентима представљеним стрелицама (средина), градијенти истог дела слике представљени бројевима [14]

Следећи корак је креирање хистограма градијената у оквиру ових 8×8 ћелија, које је илустровано на слици 8. Хистограм је низ од 9 елеменета који одговарају угловима 0°, 20°, 40°, …, 160°. Елемент низа је одабран на основу смера, а вредност елемента на основу магнитуде градијента. Градијент означен плавом бојом има угао (смер) од 80° и магнутуду 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 уписује на оба места (по 2). Ако је угао већи од 160°, он је између 160° и 180°, а како су углови од 0° и 180° еквивалентни, његова вредност би се пропорционално уписала у елементе намењен угловима од 0° и 160°. У десном делу слике је приказано како би изгледао хистограм за описани пример. Y - оса представља 0°. Како у хистограму доминирају вредности близу 0° тј. 180°, може се закључити да ова ћелија вероватно садржи ивицу јер су градијенти углавном усмерени на горе или доле.

Слика 8: Рачунање хистограма градијената (лево), график хистограма (десно) [14]

Како би дескриптор био инваријантан на промене у осветљењу, треба нормализовати хистограм. Нормализација подразумева дељење сваке вредности вектора са дужином вектора (L2 нормом). Два вектора која се разликују само у скалирању свешће се на исти вектор након нормализације. Посматрамо блок 16×16, он има 4 хистограма који када се надовежу формирају вектор од 36 елемената који се нормализuje. Прозор се онда помера за 8 пиксела и процес се понавља. Да би се израчунао коначни вектор фичера за комплетно парче слике, вектори дужине 36 се конкатенирају и креирају један велики вектор од 3780 елемената (36×105, где је број позиција на којима може да се нађе прозор 16×16). Овај вектор се сада користи за тренирање класификатора као што је SVМ. Примењују се пирамида слика и метода клизајућег прозора како би се детектовали објекти различитих величина. [14]

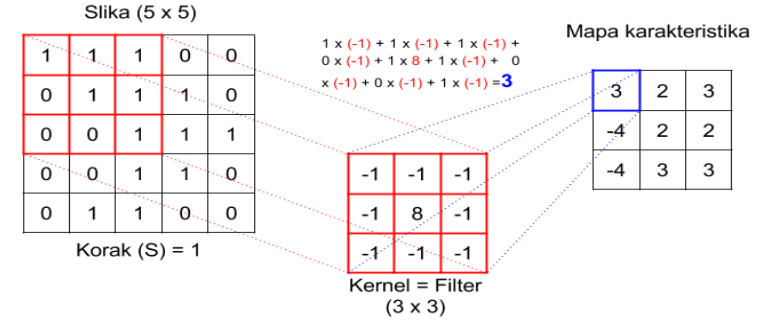
SVM је техника класификације код које је циљ пронаћи оптималну хиперраван (границу одлуке) у вишедимензионалном простору која раздваја податке који припадају различитим класама. Уколико није могуће у потпуности поделити податке, треба наћи хиперраван која максимизује маргину, односно минимизује грешке у класификацији. Маргина представља најкраће растојање од хиперравни до најближег тренинг податка који припада било којој класи. На слици 9 је лево приказано више могућих хиперравни, а десно оптимална хиперраван. Очекује се да хиперраван са већом маргином буде прецизнија приликом класификације непознатих података у односу на хиперраван са мањом маргином. Тренинг подаци који се налазе на самим маргинама се називају вектори подршке (енг. support vectors) и они се најтеже класификују, али дају највише података о самој класификацији. За случај тренинг скупа података који се не може линеарно раздвојити, проблем треба свести на проблем линеарног раздвајања, што се постиже нелинеарном трансформацијом тренинг скупа података у вишедимензионални простор где је могуће извршити линеарну поделу података. [15]

Слика 9: Могуће хиперравни (лево), оптимална хиперраван (десно) [15]

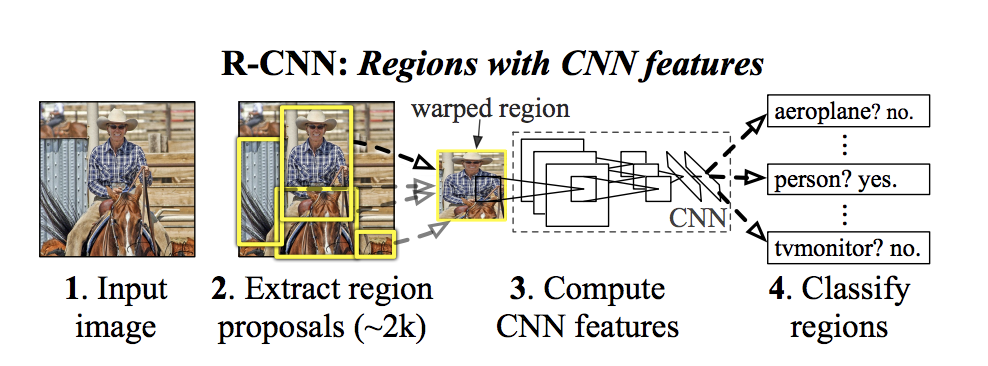
Додати неки коментар

### **R-CNN детектор**

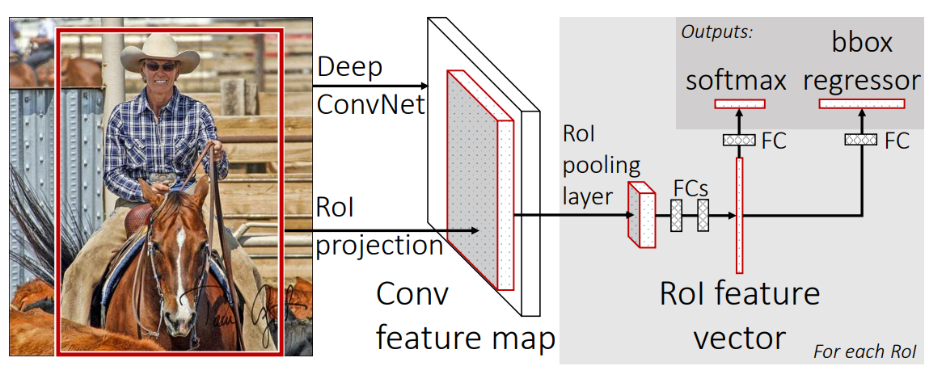
R-CNN (енг. Region-based Convolutional Neural Networks) детектор објеката из 2013. објављен у раду [16] је један од првих који примењује дубоко учење. Заснива се на конволуционим неуронским мрежама које се по својој архитектури могу сврстати у дубоке неуронске мреже и имају велику примену у домену рачунарског вида.

Конволуционе неуронске мреже су добиле име по конволуцији, оператору који се примењује у обради слика за изоштравање и замућење, као и за детектовање ивица. Конволуциони филтери (кернели) се примењују на слике како би се извукле корисне карактеристике (фичери) и креирале њихове мапе (енг. feature maps), при чему се врши и редукција резолуције слика. Филтер се представља дводимензионалном матрицом малих димензија која се састоји од реалних вредности и примењује се у конволуционом слоју мреже на слику која се обрађује у том слоју. На слици 10 може се видети примена 3х3 филтера са кораком померања 1 на монохроматску слику димензија 5х5. У сваком кораку се вредности на истој позицији множе и сабирају са производима других парова у прозору. Архитектура мреже је таква да се на почетку налази улазни слој путем кога се слика уводи у мрежу, затим следи један или више конволуционих слојева између којих се могу наћи слојеви сажимања (eng. pooling layer) који се користе са циљем прогресивног смањења слике, слој активационе функције који може значајно побољшати перформансе мреже и који се може убацити након сваког конволуционог слоја и на крају, опционо један или више потпуно повезаних (eng. Fully Connected – FC) слојева. [17]

Слика 10: Примена конволуционог филтера [17]

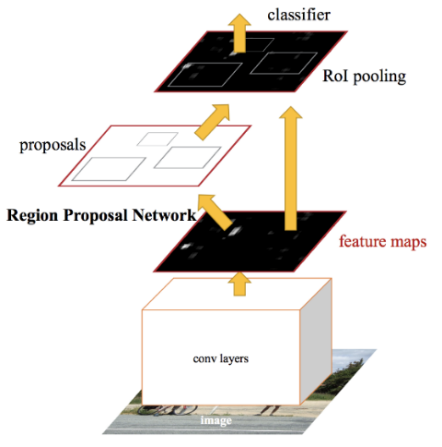
На слици 11 је приказано како модел функционише. Систем за детекцију објеката се састоји из три модула. Први модул применом селективног тражења издваја из улазне слике око 2000 предложених региона који представљају кандидате за детекцију објеката. Ови региони су у суштини правоугаоници за које постоји вероватноћа да садрже објекат (без информације о томе којој класи припада тај објекат). Алгоритам селективног тражења генерише сегменте слике који могу садржати објекат на основу боје, текстуре и облика. 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. Трећи модул је скуп SVM-а за сваку појединачну класу уз помоћ којих се класификује сваки од региона. За сваку класу се одређује score за сваки издвојени вектор фичера. Након класификације, примењује се пост-процесирање да се пречисте оквирни правоугаоници и елиминишу дупликати (bounding box regression). [16]

Слика 11: Начин функционисања R-CNN модела [16]

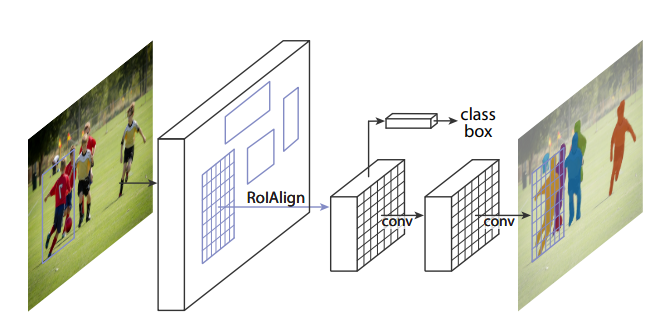
Процеси тренирања мреже и тестирања R-CNN модела трају дуго, због чега је он веома спор. Fast R-CNN из 2015. године је побољшање R-CNN метода у смислу брзине тренирања и тестирања, као и тачности детекције. На слици 12 приказана је архитектура R-CNN-а. Мрежа која као улаз има комплетну слику и скуп предложених региона процесира слику и генерише конволуциону мапу фичера. Дакле, генерише се једна мапа фичера, а не 2000. Затим, за сваки предложени регион, RoI (region of interest) pooling слој издваја вектор фичера фиксне дужине из мапе фичера за сваки од региона. Сваки фичер вектор се затим користи за класификацију региона у једну од класа и врши се побољшање тачности ориналног обухватајућег правоугаоника коришћењем bounding box regressor-а. [18]

Слика 12: Архитектура Fast RCNN-а [18]

Претходна два алгоритма користила су селективну претрагу за проналажење региона која је веома спора, односно представља уско грло у систему. Уместо овог алгоритма, Faster R-CNN користи дубоку конволуциону мрежу која предлаже регионе. На слици 13 се може видети како изгледа архитектура комплетног система. Један модул је RPN (Region Proposal Network) мрежа која на основу конволуционе мапе фичера издваја предложене регионе, а други модул на основу исте мапе фичера и предложених региона ради класификацију. Систем је јединствена неуронска мрежа за детекцију објеката. [19]



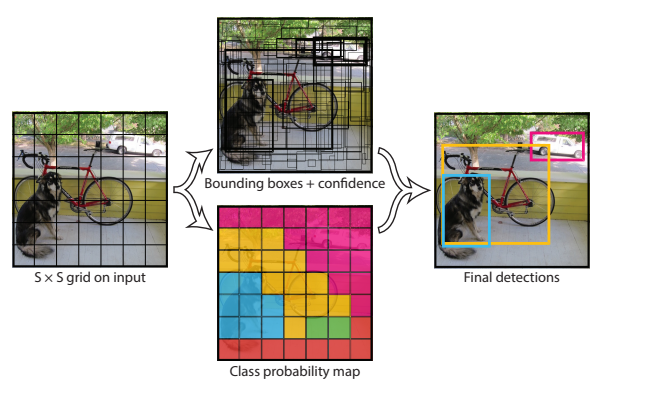
Слика 13: Faster CNN алгоритам [19]

Mask R-CNN детектор из 2017. године је унапређење Faster R-CNN и приказан је на слици 14. Разлика између њих је у томе што Mask R-CNN додаје грану за предикцију маске објекта паралелно са постојећом граном за препознавање оквирног правоугаоника. Маска која се рачуна на нивоу пиксела нам омогућава да раздвојимо објекат од позадине примењујући сегментацију инстанци (енг. instance segmentation). У овом моделу ROI Pooling модул је замењен поузданијим ROI Align модулом и додата је још једна грана из ROI Align модула. Ова додатна грана прихвата излаз ROI Align модула и прослеђује га у два конволуциона слоја, чији излаз је сама маска. [20]

Слика 14: Mask R-CNN алгоритам [20]

### **YOLO детектор**

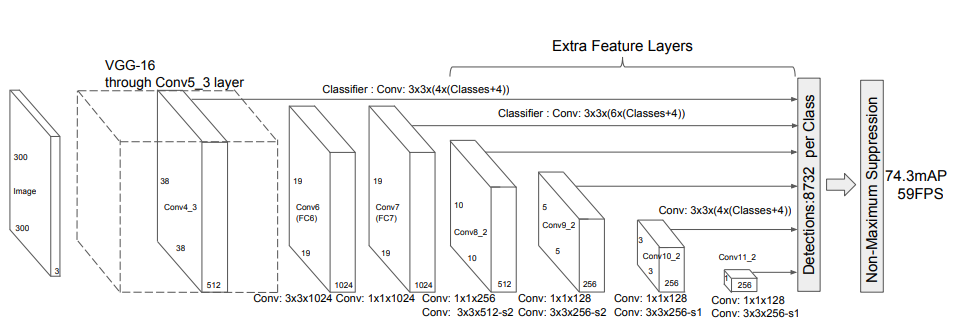
YOLO детектор је презентован у раду [21] који је објављен 2015. године и представљао је нови приступ детекцији објеката. За разлику од детектора који у основи имају класификатор и примењују детекцију над различитим деловима и величинама слике, детекција објеката овде је посматрана као проблем регресије просторно одвојених оквирних правоугаоника и њима придружених вероватноћа.

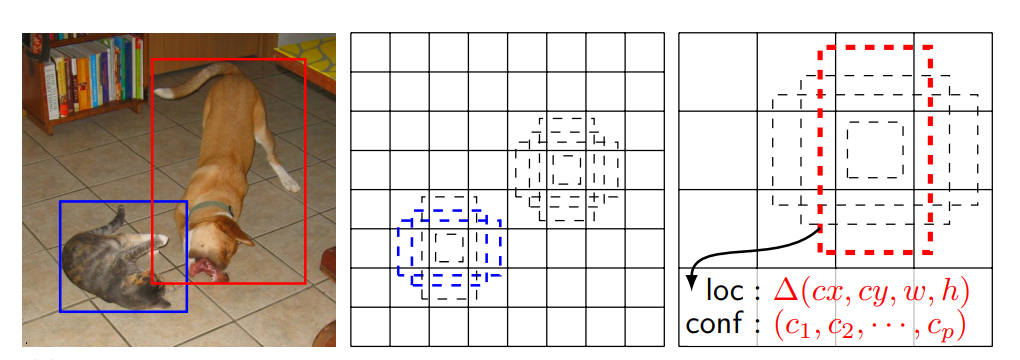
Неуронска мрежа предвиђа оквирне правоугаонике и одговарајуће вероватноће директно из комплетних слика једним погледом на слику - одатле и потиче назив детектора („погледај само једном“). Модел детектора приказан је на слици 15. Систем дели улазну слику на грид димензија S × S. Ако центар објекта упада у ћелију грида, та ћелија ће бити задужена за његову детекцију. Свака ћелија предвиђа B обухватајућих правоугаоника и поузданости Pr(Object) ∗ IOU за њих које представљају вероватноћу да правоугаоник садржи објекат. Ове поузданости казују колика је вероватноћа да правоугаоник садржи објекат. Сваки обухватајући правоугаоник има 5 предикција: x, y, w, h, и confidence (confidence score reflects how likely the box contains an object (objectness) and how accurate is the boundary box). Координате (x, y) представљају центар правоугаоника релативно у односу на границе ћелије грида. Ширина и висина (width и height) представљене су релативно у односу на целу слику. Поузданост (confidence) представља IOU[[2]](#footnote-2) (intersection over union) између предвиђеног обухватајућег правоугаоника и било ког ручно означеног правоугаоника у тест скупу слика који специфицира где се објекат налази. Свака ћелија грида такође предвиђа C условних вероватноћа за класе Pr (Classi | Object). Ове вероватноће представљају вероватноћу појављивања објеката сваке од класа у грид ћелији. Предвића се само један скуп вероватноћа по ћелији независно од броја обухватајућих правоугаоника B. Приликом тестирања множе се условне вероватноће припадања класи и индивидуалне вероватноће за сваки обухватајући правоугаоник, што нам даје вероватноћу припадања класи за сваки обухватајући правоугаоник: Pr(Classi|Object) ∗ Pr(Object) ∗ IOU = Pr(Classi) ∗ IOU. Ово нам даје вероватноћу да се та класа појављује у правоугаику, као и информацију о томе колико добро предвиђени правоугаоник представља објекат. Модел је имплементиран као конволуциона неуронска мрежа и евалуиран је на PASCAL VOC скупу података. Иницијални конволуциони слојеви издвајају фичере из слике, док потпуно повезани слојеви предвиђају излазне вероватноће и координате. Мрежа има 24 конволуционих слојева и 2 потпуно повезана слоја. Бржа варијанта мреже има мање конволуционих слојева (9) и мање филтера у тим слојевима. Како је при тренирању узето S = 7, B = 2 и C = 20 јер коришћени скуп података има 20 класа, излаз мреже је 7 × 7 ×30 тензор ((S, S, B×5 + C) = (7, 7, 2×5 + 20) = (7, 7, 30)). Архитектура је веома брза, основни модел обрађује у реалном времену 45 фрејмова у секунди на Titan X GPU, док мања верзија мреже Fast YOLO обрађује преко 155 фрејмова у секунди, задржавајући дупло већи mAP од осталих детектора који раде у реалном времену. YOLO прави више грешака у лоцирању (поготово мањих објеката), али знатно мање лажних предикција када нема ничега на слици. Обучаван је на генерализованим репрезентацијама објеката, од природнних слика до уметничких. [21]

Слика 15: Модел YOLO детектора [21]

Главни недостаци овог алгоритма су што не детектује добро мале објекте и објекте који су блиско груписани зато што дели слику на грид у коме је свака ћелија задужена за детекцију једног објекта, тако да ако постоји више малих објеката у једној ћелији они неће бити детектовани.

### **SSD детектор**

Овај метод [22] из 2016. године детектује објекте на сликама коришћењем дубоке конволуционе неуронске мреже и дизајниран је тако да ради у реалном времену. Достигао је преко 74% mAP и 59 FPS на PascalVOC скупу података. Детекција објеката обавља се у једном проласку кроз мрежу (Single Shot) и коришћена је MultiBox техника за bounding box регресију. Уводи неколико побољшања као што су мапе фичера у више величина и подразумевани правоугаоници (енг. default boxes). Конволуциона мрежа генерише неколико фиксних подразумеваних оквирних правоугаоника различитих величина и предикције (вероватноће и помераје правоугаоника) за присуство објеката у тим превоугаоницима коришћењем малих конволуционих филтера примењених на мапе фичера, што је праћено non-maximum suppression како би се добиле коначне предикције. Детекција објеката се састоји из два корака: издвајање мапа фичера и примена конволуционих филтера за детекцију објеката. Архитектура мреже приказана на слици 16 састоји се из два дела и заснована је на VGG-16 мрежи која се користи за класификацију слика. VGG-16 мрежа је одсечена пре класификационог слоја и њени потпуно повезани слојеви 6 и 7 су конвертовани у конволуционе слојеве. Затим следи скуп помоћних 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. SSD за издвајање мапа фичера користи VGG16, али могу се користити и други модели као основа.

Посматрајмо слој Conv4\_3. Слика се дели на 38 × 38 ћелија, при чему се за сваку од њих генеришу 4 предикције за сваки од подразумеваних правоугаоника, тако да имамо укупно 38 × 38 × 4. Генерисање више предикција се зове multibox. Предикција се састоји од оквирног правоугаоника и 21 вредности које означавају припадност свакој од класа (20 класа за објекте и једна класа за позадину, тј. случај кад нема објекта). Рачунање предикција се обавља применом 3 × 3 конволуционог филтера над сваком ћелијом. За класу уоквиреног објекта узима се класа са највећом вредношћу score-а. Овај поступак се обавља у сваком конволуционом слоју, при чему слојеви имају мапе фичера различитих димензија. Резолуција мапа фичера се смањује како се смањује величина слојева, што омогућава детекцију објеката различитих величина и значајно повољшава тачност детектора. Укупно има 6 конволуционих слојева, при чему се у неким од њих генеришу 4, а у неким 6 предикција. SSD укупно генерише 8732 (38\*38\*4 + 19\*19\*6 + 10\*10\*6 + 5\*5\*6 + 3\*3\*4 + 1\*1\*4 = 8732) предикција. Објекти могу бити различитих облика и потребно је да подразумевани правоугаоници покрију што више њих како би детектор могао да детектује што више типова објеката. Ови подразумевани правоугаоници су ручно одабрани и тежило се томе да их буде што мање како би цео процес био бржи. Као што је речено, за сваки подразумевани правоугаоник се генерише једна предикција. Предиктован оквирни правоугаоник је дефинисан релативно у односу на подразумевани правоугаоник коришћењем 4 вредности: (∆cx, ∆cy, ∆w, ∆h) које представљају офсете у односу на центар (cx, cy), ширину и висину подразумеваног правоугаоника. На нивоу једног feature map слоја користе се исти поразумевани правоуганици за сваку ћелију и они су центрирани у тој ћелији, али различити слојеви користе различите подразумеване правоуганике услед разлике у резолуцији. SSD дефиниче фактор скалирања за сваки од слојева. Гледано са лева десно, фактор скалирања иде од 0.2 (или 0.1 некад) до 0.9. Комбиновањем ових фактора скалирања и жељених aspect ratios израчуванају се ширина и висина подразумеваних правоугаоника. Предикције се могу класификовати као позитивна и негативна поклапања, при чему се само позитивна користе за рачунање localization cost (the mismatch of the boundary box). Ако одговарајући подразумевани правоугаоник има IoU већи од 0.5 са ground truth, поклапање се сматра позитивним, а у супротном негативним. ground truth правоугаоници су они којима су ручно означени објекти на тренинг сликама. На пример, на слици имамо два подразумевана правоугаоника поклопљена са мачком и један са псом, који се сматрају позитивним преклапањима, а остали правоугаоници се сматрају негативним. На слици 17 је приказан и пример како SSD комбинује мапе фичера различитих величина и подразумеване оквирне правоугаонике да детектује објекте различитих величина и аspect ratios. Пас одговара једном подразумеваном правоугаонику (црвеном) у слоју са 4×4 мапом фичера, али ниједном подразуменом 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). The localization loss је разлика у поклапању између ground truth правоугаоника и предиктованог оквирног правоугаоника, при чему се посматрају само позитивна поклапања. При рачунању се користи L2-Norm. The confidence loss је губитак услед предикције класе, при чему се посматрају и позитивна и негативна поклапања.g to the confidence score of the corresponding class (негативна поклапања се односе на позадинску класу). За рачунање се користи Categorical cross-entropy. Финална loss function се израчунава као тежинска сума localization loss и confidence loss. Како се генерише много више предикција него што је присутно објеката, има много више негативних него позитивних поклапања, што креира дисбаланс међу класама и утиче на процес тренирања. Уместо коришћења свих негативних поклапања, SSD, она се сортирају по calculated confidence loss и бирају се она са највећим губитком, при чему се одржава однос између позитивних и негативних на 3:1. Још једна важна ствар за побољшање тачности је повећање скупа података (енг. data augmentation) коришћењем flipping, cropping, and color distortion, а како би се обухватили објекти различитих величина и облика, свака тренинг слика може бити насумично узета као оригинал, као насумично парче или парче са IoU of 0.1, 0.3, 0.5, 0.7 или 0.9. The sampled patch will have an aspect ratio between 1/2 and 2. Then it is resized to a fixed size and we flip one-half of the training data. In addition, we can apply photo distortions. На крају, користи се non-maximum како би се уклониле предикције дупликати. Предикције се сортирају по поузданости и задржава се најбољих 200 предикција по слици, док се елиминишу предикције са поузданошћу мањом од 0.01 и IoU мањим од 0.45. SSD даје слабије резултате за мале објекте зато што они могу бити детектовани само у слојевима највеће резолуције, а они садрже мање значајне фичере. Тачност детекције расте са повећањем броја подразумеваних правоугаоника, али по цену брзине. [23]

Слика 16: Архитектура SSD детектора [22]

Слика 17: [22]

## **Препознавање лица**

Када је реч о препознавању лица прво треба дефинисати појам детекције лица зато што је то процес који најчешће претходи препознавању. Детекција лица је рачунарска техника која се користи за проналажење људског лица на дигиталним сликама и представља први корак било које обраде лица. Може се сматрати спефицичним случајем детекције објеката и односи се на испитивање да ли се на слици налази лице, где је лоцирано и које је величине. [24] Као најпознатији алгоритми који се користе за детекцију лица могу се издвојити Viola-Jones и HOG + Linear SVM (описани у претходном поглављу), Eigenfaces и алгоритми засновани 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), комплексност позадине (присуство великог броја објеката) итд. Након детекције лица могу се издвојити карактеристичне тачке лица које означавају истакнуте деловe лица попут очију, обрва, носа, усана и образа. Карактеристичне тачке се успешно примењују нпр. за поравнање лица, замену лица, препознавање лица, одређивање положаја главе, праћење покрета очију.

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. [25] 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). Препознавању најчешће претходи детекција лица како би се обрађивао само део слике који представља лице. 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. Неке методе за поравнање лица посматрају 3D модел лица, а једноставније методе се ослањају само на карактеристичне тачке (конкретно, на тачке које одређују очи) да би извршиле трансформације над сликом попут ротације, транслације и скалирања.

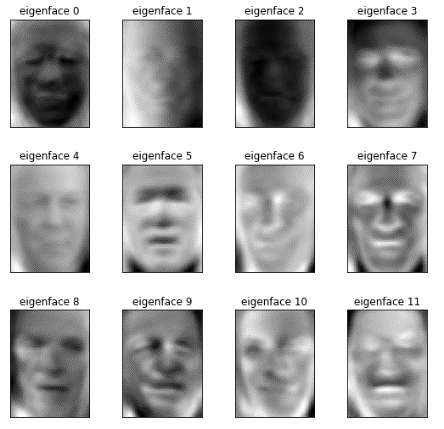
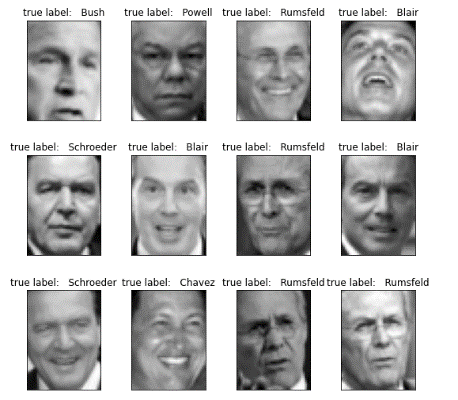
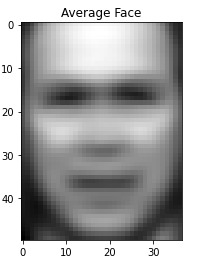
Код система за препознавања лица може бити велики проблем уколико неко злонамерно покуша да приступи систему представљајући се као неко други. Како се препознавање врши на основу података са камере, може се десити да неко подметне слику или видео друге особе и, уколико нема провере да ли је лице на камери реално, систем подметнута лица може препознати као валидна. Постоје различити приступи овом проблему, а неки од њих су анализа текстура (рачунање Local Binary Patterns над регионом лица и коришћење SVM за класификацију реалних и лажних лица), анализа фреквенци (Фуријеов домен лица), анализа променљивих величина (попут вредности пиксела) 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 претходно наведених приступа. [26]

Прво 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 све више се користе алгоритми дубоког учења. Један од најзначајнијих алгоритама за развој препознавања лица је Eigenfaces, где је коришћена техника линеарне алгебре за редукцију димензија која се зове Principal Component Analysis (PCA). Затим је објављена метода Fisherfaces која користи Linear Discriminant Analysis (LDA). Појавила се и метода базирана на фичерима Local Binary Patterns (LBP) која се и данас користи у многим апликацијама. Када је у питању дубоко учење, постоје специјалне архитектуре које се зову siamese networks. FaceNet и OpenFace су једни од најпопуларнијих модела дубоког учења који се користе за препознавање лица. [27]

### **Eigenfaces**

У раду [28] из 1991. године представљен је Eigenfaces алгоритам за детекцију и препознавање лица који се заснива на редукцији димензија простора слика лица коришћењем PCA (енг. Principal Component Analysis).

Алгоритам најпре обрађује скуп од М тренинг слика лица које су сиве, исте величине и поравнате тако да се кључни делови лица што више поклапају. Посматрамо слике димензија N×N (уопштем случају могу бити произвољне величине) које се могу линеаризовати и представити као вектор димензија N2, односно као тачка у простору са N2 димензија. Израчунава се средње лице које је вектор дужине N2 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 заправо померамо средњу тачку (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 N2×N2 иможе се израчунати множењем матрице А и АT, при чему је матрица А добијена смештањем вектора тренинг слика у колоне. У пресеку неке врсте и колоне 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 N2 и може се посматрати као лице које личи на духа и које називамо „еigenface“ или „својствено лице“. Свако лице из тренинг скупа може бити представљено као линеарна комбинација К својствених лица која најбоље репрезентују варијације међу лицима и која описују К-димензиони простор који називамо „простор лица“, при чему је К 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 неколико eigenface репрезентација. Светлији региони одговарају већим варијацијама, а тамнији мањим.



Слика 18: Тренинг лица (лево), средње лице (средина), еigenfaces (десно) [29]

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 К-димензиони простор као линеарна комбинација својствених лица. За потребе детекције лица, из К-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 К-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. [29]

Предност Eigenfaces алгоритма је што није рачунски захтеван и брзо се извршава Један од недостатака овог алгоритма је што захтева поравнање лица приликом тренирања и препознавања зато што се ради на нивоу пиксела, тако да је потребно да се карактеристике лица скоро савршено поклапају. Такође, захтева фронтална лица.

### **Fisherfaces**

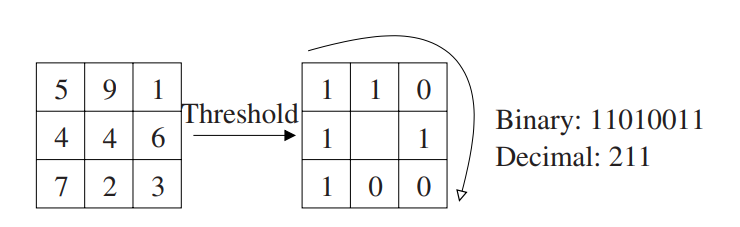
 Fisherfaces метод представљен 1997. године у раду [30] је уз Eigenfaces један од најпопуларнијих алгоритама препознавања лица. Овај алгоритам се такође заснива на редукцији димензија. Eigenfaces алгоритам користи анализу главних компоненти (PCA), док Fisherfaces користи линеарну дискриминантну анализу (LDA).

Слика 19: Слике лица при различитом осветљењу [30]

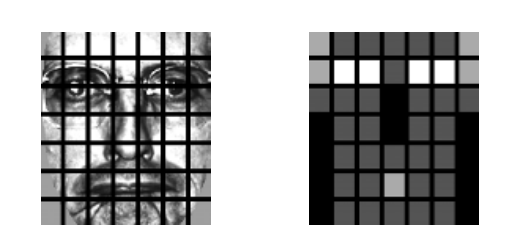
Сваки пиксел на слици посматран је као тачка у високодимензионом простору. Слика је линеарно пројектована у нискодимензиони потпростор који није осетљив на промене у осветљењу и експресији лица. Eigenfaces користи пројекције које максимизују разлике између лица у целом скупу слика, што не даје добре резултате при препознавању када постоје промене у осветљењу јер су веће разлике између слика истог лица при различитом осветљењу него између слика на којима су лица различитог идентитета. Fisherfaces користи пројекцију која ће максимизовати однос расподеле између класа и расподеле унутар класа. Тежи се максимизацији удаљености лица различиитх класа, а минимизацији удаљености унутар сваке класе. Израчунају се матрице коваринсе између класа и унутар класа, а затим eigen вектори и еigen вредности. Бирају се eigen вектори са највећом eigen вредношћу. Као код Eigenfaces алгоритма, тренинг слике се пројектују у потпростор слика, улазна слика се такође пројектује у овај потпростор и упоређују се растојања. [30]

Метод је дао боље резултате од Eigenfaces алгоритма 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 алгоритма је што не подржава промене у пози и, слично Eigenfaces алгоритму, захтева да слике лица буду поравнате.

### **LBP**

Алгоритам који користи Local Binary Patterns (LBP) за препознавање лица приказан је у раду [31] који је објављен 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 (слика 20). Затим се хистограм ових вредности користи као дескриптор текстуре.

Слика 20: LBP

Први корак у алгоритму 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, хистограм је 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 за сваку ћелију. Бели квадрати (очи) имају тежину 4 (њихови хистограми су помножени са 4), светло сиви тежину 2 (уста и уши), тамно сиви 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, конкатенирају се) и затим се примени k-NN (k=1) са растојањима како би се нашло најближе лице из тренинг података. [31]

Слика 21: Лево: Оригинална слика лица. Десно: тежинска шема за 7×7 ћелија [31]

Овај алгоритам може да се ажурира приликом додавања новог лица у систем, док неки други 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 алгоритам. Ефикасан је и омогућава веома брзо издвајање фичера.

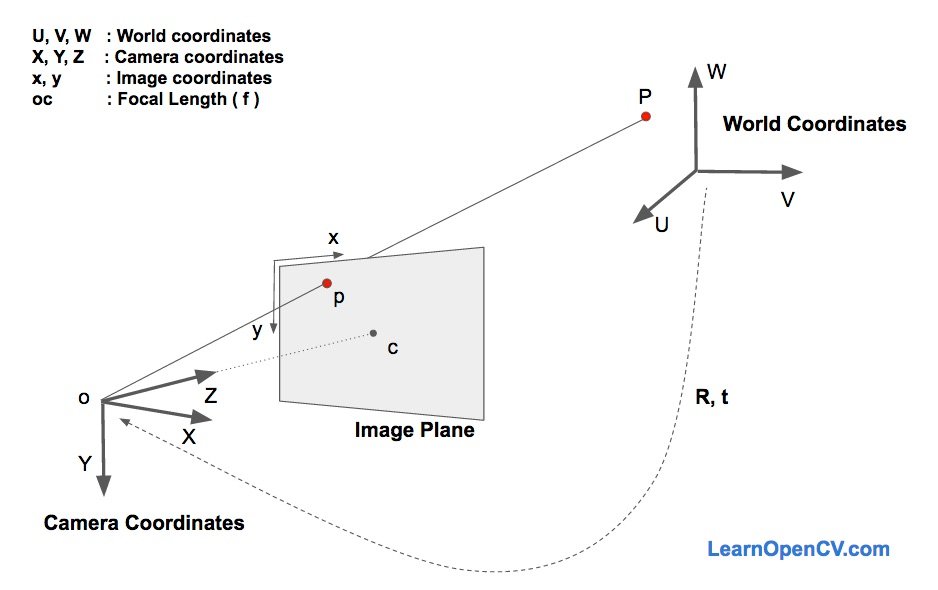
### **Дубоко учење**

Дубоке конволуционе неуронске мреже могу се користити за препознавање лица са великом тачношћу. 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). Тренирање се обавља тако што се користе 3 различите слике (triplets), при чему две припадају истој особи, а трећа некој другој особи. Неуронска мрежа генерише 128-димензионе векторе за сваку од слика и 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 слике да би мрежа научила да поуздано генерише мере за сваку особу. Мере генерисане за различите слике исте особе треба да буду скоро исте.

Слика 22: 128 мера лица [32]

Када се мрежа једном истренира може се користити за генерисање вектора за било које лице, па и потпуно непознато. Да би се непознато лице препознало треба наћи у бази познатих лица оно које има најближе мере са непознатим, што се може постићи коришћењем класификатора. [32]

## **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. Поза објекта се одређује када имамо калибрисану камеру и знамо локације n 3D тачака које одговарају 2D пројекцијама на слици. Објекат се може кретати у односу на камеру применом транслације или ротације, а то кретање можемо посматрати и као кретање камере око објекта. Транслацијом се назива померање камере са њене тренутне 3D позиције (X, Y, Z) на нову 3D позицију (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 на више начина: коришћењем Ојлерових углова, 3x3 ротационе матрице или смером ротације и углом. За одређивање позе главе потребно је знати 2D координате (x,y) неколико тачака на слици (врх носа, врх браде, углови очију и углови усана), 3D координате истих тачака, као и унутрашњи параметри камере (фокална дужина камере, оптички центар слике и параметри радијалне дисторзије). 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. Ми посматрамо 3 координатна система: светски координатни систем, координатни систем камере и координатни систем слике. 3D координате неколико тачака лица представљене су у светским координатама и ако знамо ротацију и транслацију можемо их 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.

Слика 23: Координатни системи [33]

На слици 23 о је центар камере, а раван која је приказана је раван слике. Нас интересују једначине које дају пројекцију p3D тачке P на раван слике.Нека је (U, V, W) позиција тачке P у светским координатама. Ако знамо ротациону матрицу R (3x3) и транслациони вектор *t* (3x1), можемо израчунати (X, Y, Z) координате тачке *P* у координатном систему камере на следећи начин:

,

односно:

.

Ако имамо довољан број парова (X, Y, Z) и (U, V, W), можемо решити систем линеарних једначина и добити непознате rij и (tx, ty, tz). Нама су познате тачке у 3D моделу (U, V, W), али не знамо (X, Y, Z), већ само 2D тачке (x, y). У одсуству радијалне дисторзије координате (x, y) тачке pдате су једначином:

где су fx и fyфокалне дужине у x и y смеровима, а (cx, cy) је оптички центар. Непознати фактор скалирања *s* постоји у једначини услед чињенице да не знамо дубину слике. Ако спојимо било коју 3D тачку P са центром камере, тачка p, у којој зрак пресеца раван слике је пројекција тачке P. Све тачке дуж зрака када се споје са центром камере имаће исту пројекцију на раван слике, тако да коришћењем претходне једначине можемо добити само (X, Y, Z) са скалирањем s. Сада наша једначина изгледа овако:

и може се решити коришћењем метода који се зове директна линеарна трансформација [(DLT)](https://en.wikipedia.org/wiki/Direct_linear_transformation). Овај метод се може користити кад год имамо проблем где је једначина скоро па линеарна, али је помножена непознатим фактором. Метод није баш поуздан зато што ротациона матрица R има три степена слободе, а матрица која се појављује у DLT решењу има 9 бројева. DLT решење не минимизује the correct objective function. У идеалном случају, ми желимо да минимизујемо **грешку пројекције**. Ако знамо тачну позу (R и t), можемо предвидети 2D локације 3D тачака са слике пројектовањем 3D тачака на 2D слику. Како знамо 2D карактеристичне тачке лица, можемо посматрати растојање између пројектованих 3D тачака и 2D карактеристичних тачака лица. Када процењена поза буде савршена, 3D тачке пројектоване на слику ће се поклопити са 2D карактеристичним тачкама. Када је процењена поза нетачна, можемо израчунати **грешку пројекције** између пројектованих 3D тачака и 2D карактеристичних тачака лица. Овај метод се може побољшати итеративном променом вредности R и t тако да се грешка пројекције смањује: [Levenberg-Marquardt](https://en.wikipedia.org/wiki/Levenberg%E2%80%93Marquardt_algorithm) оптимизација. OpenCV библиотека нуди функције које решавају PnP проблем и могу се користити за одређивање позе. [33]

## **Праћење покрета очију**

За праћење покрета очију и усана могу се искористити карактеристичне тачке лица.

## **Детекција причања**

1. Haar-ов талас је секвенца функција квадратног облика које формирају фамилију таласа или базу. [12] [↑](#footnote-ref-1)
2. IoU (пресек над унијом) је метрика за евалуацију тачности детектора објеката. Представња количник површине преклапања предиктованог правоугаоника и ручно означеног правоугаоника из тест скупа података и површине уније та два правоугаоника. Добром предикцијом се сматра IoU већи од 0,5. [↑](#footnote-ref-2)