



УНИВЕРЗИТЕТ У НИШУ  
ЕЛЕКТРОНСКИ ФАКУЛТЕТ  
Катедра за рачунарство



Никола Петровић

**ПРЕПОЗНАВАЊЕ СИГНАЛА ПОМОЋНОГ  
ФУДБАЛСКОГ СУДИЈЕ**

- ДИПЛОМСКИ РАД -

Ниш, 2022.





УНИВЕРЗИТЕТ У НИШУ  
ЕЛЕКТРОНСКИ ФАКУЛТЕТ  
Катедра за рачунарство



# ПРЕПОЗНАВАЊЕ СИГНАЛА ПОМОЋНОГ ФУДБАЛСКОГ СУДИЈЕ

## - ДИПЛОМСКИ РАД -

### **Задатак:**

Упознати се са техникама за детекцију позе тела, тј. карактеристичних тачака на телу. Проучити постојеће библиотеке које пружају ову функционалност. Развити и имплементирати алгоритам који на основу овако добијених тачака врши препознавање сигнала помоћног фудбалског судије. У практичном делу имплементирати прототип апликације која обезбеђује препознавање основних сигнала које помоћне судије користе у току фудбалских утакмица.

**Ментор:** проф. др Александар Милосављевић    **Кандидат:** Никола Петровић 17344

## Комисија:

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

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

Датум предаје: \_\_\_\_\_

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

Датум предаје: \_\_\_\_\_

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

Датум одбране: \_\_\_\_\_

Ниш, 2022.

## Садржај

<b>1. УВОД</b>	<b>4</b>
<b>2. ДЕТЕКЦИЈА ПОЗЕ ТЕЛА НА ОСНОВУ СЛИКЕ</b>	<b>6</b>
<b>2.1. Класичан и приступ препознавања позе тела заснован на дубоком учењу</b>	<b>8</b>
2.1.1. Класични приступ	8
2.1.2. Приступ заснован на дубоком учењу	8
<b>2.2. Категорије метода за препознавање позе тела засноване на дубоком учењу</b>	<b>9</b>
2.2.1. Generative и Discriminative	10
2.2.2. Top-down и Bottom-up	11
2.2.3. Regression-based и Detection-based	12
2.2.4. One-stage и Multi-stage	13
<b>2.3. Библиотеке које омогућавају детекцију позе тела на основу слике</b>	<b>14</b>
2.3.1. OpenPose	14
2.3.2. AlphaPose	19
2.3.3. Detectron	21
<b>2.4. Поређење библиотека за детекцију позе тела</b>	<b>23</b>
<b>2.5. MediaPipe</b>	<b>26</b>
<b>2.6. Зашто MediaPipe?</b>	<b>31</b>
<b>3. ПРЕПОЗНАВАЊЕ СИГНАЛА ФУДБАЛСКОГ СУДИЈЕ ПОМОЋНИКА</b>	<b>34</b>
<b>3.1. Идеја</b>	<b>36</b>
<b>3.2. Класификација сигнала фудбалског судије помоћника</b>	<b>39</b>

<b>4. ИМПЛЕМЕНТАЦИЈА ДЕМО АПЛИКАЦИЈЕ ЗА ПРЕПОЗНАВАЊЕ СИГНАЛА ФУДБАЛСКОГ СУДИЈЕ ПОМОЋНИКА</b>	<b>45</b>
<b>5. ЗАКЉУЧАК</b>	<b>62</b>
<b>Литература</b>	<b>64</b>

# 1. УВОД

Фудбал, као најбитнија споредна ствар на свету, представља спорт који је из године у годину све популарнији, граде се стадиони са све већим капацитетима, изучавају се нове тактике игре, они који се професионално баве фудбалом су планетарно популарни и многи од њих су идоли младима, трансфери играча и тренера достижу огромне новчане суме. Фудбалом се бави велики број људи на планети и многи од њих су задужени да побољшају квалитет игре и повећају њену популарност, све са циљем постизања спектакуларног и узбудљивог доживљаја код навијача. Ова игра у људима буди емоције, доноси навијачима срећу, радост, тугу, лутњу или их некада изненади. У историји је чак забележен и један рат чији почетак је иницирала једна фудбалска утакмица, то је био рат између две државе Средње Америке, Салвадора и Хондураса, 1969. године, након тромеча за одређивање екипе која ће учествовати на светском првенству, рат је назван „Фудбалски рат“.

Квалитет фудбала се побољшава као резултат разних подухвата, од улагања у играче и у њихов физички и ментални напредак, преко изградње квалитетне инфраструктуре, ваљаног представљања у медијима и доброг маркетинга, па све до исправног примењивања правила фудбалске игре (ПФИ). Што нас доводи до закључка да нема добrog и квалитетног фудбала без судија који су добро обучени, познају правила фудбалске игре и физички су спремни да испуне захтеве и темпо играча на највишем светском нивоу.

Фудбалски судија је особа која на фудбалским утакмицама примењује правила фудбалске игре, чија једна погрешна одлука може довести до огромних новчаних губитака, затим до нездовољства самих играча, а онда и до нездовољства огромног аудиторијума широм планете, што негативно утиче на популарност фудбала, и у неретком броју случајева доводи до конфронтација и физичких сукоба на терену и ван њега. Поред ове главне улоге, посао фудбалских судија јесте и то да дају пример понашања играчима, да буду ауторитативни, разумни, сналажљиви и присебни у сваком тренутку, али и да лепо изгледају и да свима на терену и ван њега буде евидентно и јасно коју су одлуку донели, што подразумева недвосмислену и свима добро познату сигнализацију прекршаја на терену. Таква сигнализација доприноси лепоти фудбала и помаже у томе да не дође до конфузије код играча и same публике.

Постоје технологије које судијама помажу у доношењу одлука, најпопуларније су GLT (Goal Line Technology [1]) и VAR (Video Assistant Referee [2] ). GLT служи за доношење одлуке о томе да ли је лопта прешла линију врата у деликатним ситуацијама, док VAR служи за преглед успореног видео снимка ради доношења исправне одлуке у ситуацијама где постоје индиције да је судија пропустио да досуди неки споран прекршај.

Најбитнија сигнализација јесте сигнализација судија помоћника, где судија помоћник покретима свога тела да судији и комплетном аудиторијуму треба дати до знања да је дошло до одређене врсте прекршаја и исправно сигнализира тај прекршај.

На први поглед то делује једноставно, међутим, различити прекршаји захтевају врло сличне покрете судије помоћника, док с друге стране, судија помоћник при показивању мора да покаже озбиљност и сигурност у своју одлуку, а то подразумева исправан став, исправан положај руку, ногу, тела, главе и шаке. Да тиме допринесе лепоти фудбала и ужитку целокупног аудиторијума који прати тај фудбалски меч. Постоје људи који оцењују рад судија и судија помоћника, тиме закључујемо и да позиција, поштовање и поверење судије помоћнику у струци, између осталог, зависи и од исправности његовог сигнализирања прекршаја.

Идеја је да се уз помоћ рачунарског вида помогне судијама помоћницима у тренирању сигнализирања, тако што ће њихови покрети и положаји тела бити обрађени и на основу истих бити установљено шта је судија сигнализирао. То је могуће због тога што се математички може одредити да ли одређени део тела човека задовољава услове за правилно сигнализирање одређеног прекршаја фудбалске игре. Комбинацијом исправних положаја делова тела од интереса, где су сви ти делови тела међусобно зависни, може се препознати конкретан сигнал судије помоћника.

Рачунарски вид пружа широк скуп библиотека и метода за препознавање позе тела човека, у раду је извршена анализа ових библиотека и процена која би библиотека била најбоља за решавање овакве врсте проблема. Такође, описани су алгоритми на којима се заснивају саме библиотеке за препознавање позе тела и извршено поређење ефикасности рада истих. Проблем је решен у програмском језику Python коришћењем OpenCV библиотеке и MediaPipe cross-platform која између осталог пружа и могућност препознавања позе тела код човека. У раду су детаљно описане све предности које MediaPipe има у односу на најпопуларнију библиотеку за препознавање позе тела OpenPose, које су допринеле одабиру ове технологије за конкретан проблем.

Рад се састоји од пет поглавља, увода који описује настанак идеје, затим теоријског дела у коме се налазе резултати истраживања разних библиотека, поређења функционалности метода у односу на одређене параметре, начина рада технологија за препознавање позе тела и објашњење због чега је изабран MediaPipe. Након теоријског дела описано је која логика се крије иза решавања проблема препознавања сигнала судије помоћника, посвећено је поглавље имплементацији саме демо апликације и на крају је изнет закључак у коме се врши осврт на рад, предности и могуће недостатке рада, као и на могућа унапређења демо апликације.

## 2. Детекција позе тела на основу слике

Детекција позе тела или Human Pose Estimation (HPE) представља део рачунарског вида чији је задатак да на основу улазне слике или видео записа препозна кључне тачке на телу човека (зглобови, очи, нос, уста, глава..) и на основу истих процени позу у којој се налази.

Ова област је са становишта примене, али и изучавања, веома интересантна, тренутно је најкоришћенија у фитнес и спортским апликацијама где се процењује исправност извршавања вежби и држања тела, такође има широку примену у видео играма и прављењу уверљивих анимација стварних актера, као што су потези NBA играча или фудбалера.

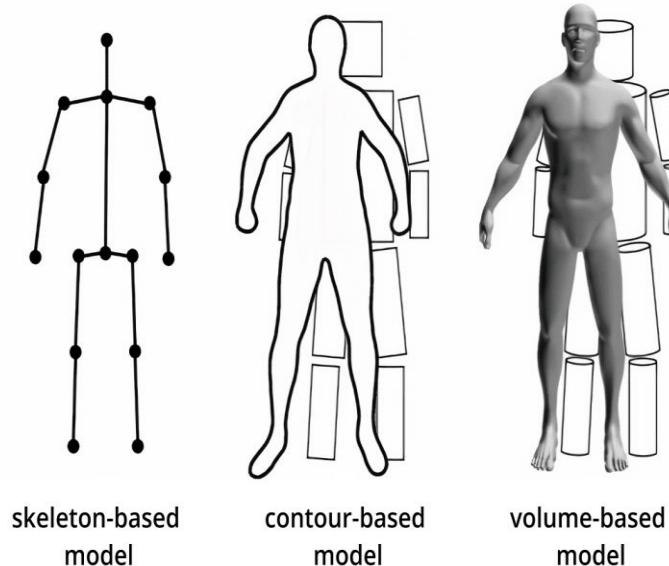
Улазни подаци који се користе могу бити RGB фотографије, у највећем броју случајева, из простог разлога што је највише доступних камера управо RGB, затим Depth Image, којом се RGB слика може допунити и добити прецизнији и сложенији модел, а вредност пиксела се односи на удаљеност од камере мерено временом лета. И на крају најмање популарни вид улазних фотографија, Infra-red, где се вредност пиксела односи на количину инфрапед светла које се рефлектује назад у камеру.

У основи ова област се своди на идентификовање тачака које могу описати позу особе у простору преко x, y или z координата, разлика је у томе сто се две координате користе за 2D представљање модела, док се помоћу три координате модел представља у 3D простору. Наравно, 3D препознавање позе је теже остварити, али су резултати немерљиво бољи и могуће је извршити предвиђање просторног позиционирања особе.

Постоје три врсте приступа за моделирање људског тела на основу препознатих тачака на телу [3]:

1. Skeleton-based модел
2. Contour-based модел
3. Volume-based модел

## HUMAN BODY MODELS



*Слика 1.1. Модели за представљање људског тела*

Skeleton-based – модел заснован на скелету, познат и као кинематички модел представља скуп моћусобно повезаних тачака, чија презентација представља структуру скелета људског тела. Овај модел се може описати и као граф, где тачке представљају чворове, а линије између тачака везе између елемената графа. Ова технологија је веома једноставна и флексибилна, због тога има широку примену и у 2D и 3D препознавању позе тела. Њен недостатак јесте недостатак информација о ширини и контурама људског тела, тај проблем се решава у наредним приступима.

Contour-based – планарни модел или модел заснован на контурама, користи се за представљање изгледа и облика људског тела, обично су делови тела представљани вишеструким правоугаоницима који приближно описују контуру људског тела

Volume-based – модел заснован на запремини, људско тело се представља у облику геометријских тела као што су: цилиндри, купе.. али и у виду мреже, то се постиже скенирањем уз помоћ 3D скенера и све је популарнији вид представљања људског тела код препознавања позе тела.

## **2.1. Класичан и приступ препознавања позе тела заснован на дубоком учењу**

Ови приступи се користе како би се геометријски лакше разумело кретање људског тела, класични приступ се користио годинама уназад, међутим у садашњости је приступ заснован на дубоком учењу (енг. Deep Learning), односно коришћењу конвулуционах неуронских мрежа (енг. CNN – Convolutional Neural Network), много заступљенији и ефикаснији.

### **2.1.1. Класични приступ**

Овај приступ углавном тежи ка примени техника које подразумевају алгоритме за машинско учење. Користи се PSF (енг. Pictorial Structure Framework) како би се предвиделе кључне тачке на људском телу, на основу већ постојећих модела људских делова тела препознаје делове тела на непознатим фотографијама, он садржи две компоненте: Discriminator и Prior.

- **Discriminator** претпоставља да је одређени део тела присутан на одређеној локацији на слици,
- **Prior** служи за рачунање вероватноће да је резултат који даје discriminator исправан на основу већ постојећих поза

ово нас доводи до закључка да овај приступ добро ради када су на слици сви делови тела у већ познатом положају и када нема „преплитања“ делова тела тј. када неки део тела не прекрива други.

Да би се превазишли ови проблеми коришћене су помоћне методе, међутим и поред додатних метода оваквом приступу је недостајала тачност, корелација и генерализација, тако да је било питање времена када ће се усвојити бољи приступ препознавања позе тела.

### **2.1.2. Приступ заснован на дубоком учењу**

Приступи засновани на дубоком учењу су дефинисани својом способношћу да генерализују било коју функцију.

Када су у питању задаци компјутерског вида, конвулуционе неуронске мреже (CNN) у великој мери надмашују све друге алгоритме, што важи и за HPE (детекцију положаја тела код човека).

CNN има могућност да издвоји обрасце и информације из улазне слике са више прецизности и тачности него било који алгоритам, што га чини веома корисним за задатке као сто су класификација, детекција и сегментација.

За разлику од класичног приступа, где су информације и обрасци били мануелно утврени на подаци, CNN може научити сложене шаблоне и са много већом тачношћу представити информације о сликама које требају да се обраде, наравно уз доступност довољне количине података.

## **2.2. Категорије метода за препознавање позе тела засноване на дубоком учењу**

Све методе за препознавање позе тела имају исти задатак, међутим, свака користи специфичне карактеристике слике како би препознала људско тело. Постоји осам различитих врста приступа и свака има себи супротну у односу на ресурсе и информације које користи [4].

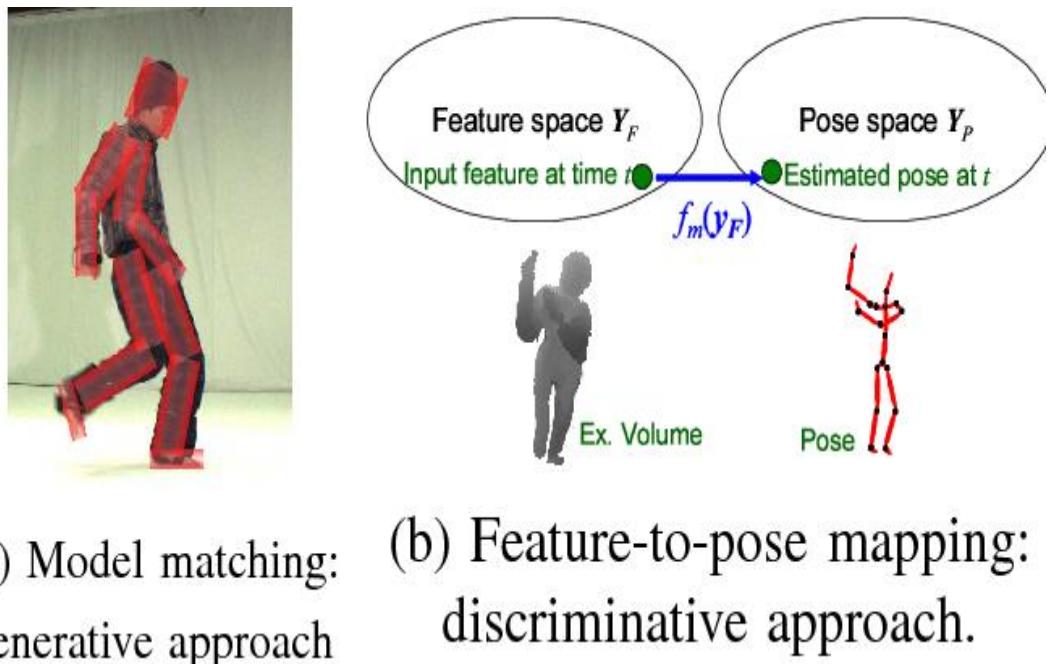
1. Generative (енг. human body model-based)
2. Discriminative (енг. human body model-free)
3. Top-down (енг. from high level abstraction to low-level pixel evidence)
4. Bottom-up (енг. from low-level pixel evidence to high-level abstraction)
5. Regression-based (енг. directly mapping from input images to body joint positions)
6. Detection-based (енг. generating intermediate image patches or heatmaps of joint locations)
7. One-stage (енг. end-to-end training)
8. Multi-stage (енг. stage-by-stage training)

### 2.2.1. Generative и Discriminative

Главна разлика између ових метода јесте у томе да ли метода користи моделе људског тела или не.

Generative метода користи моделе људског тела и у зависности од различитих репрезентација модела људског тела може се заснивати на препознавању постојећих модела, затим геометријске пројекције из различитих погледа на 2D или 3D простор, док с друге стране discriminative методе директно уче мапирање са улазних података или траже податке у већ постојећим примерима, без коришћења постојећих модела људског тела.

Discriminative методе су углавном брже од generative, међутим долази до проблема када требају да се препознају позе за које не постоје утренирани подаци.



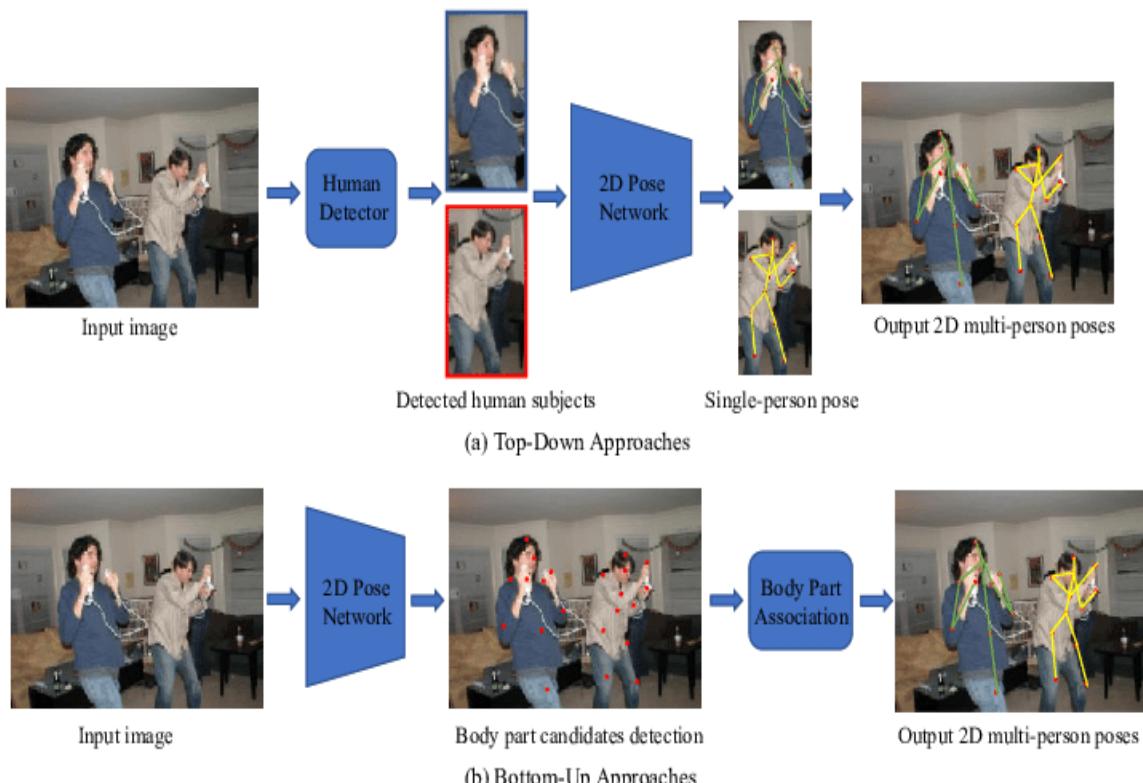
Слика 2.1. Генеративни и дискриминативни приступ

## 2.2.2. Top-down и Bottom-up

Ове методе се користе за препознавање више особа на слици, могу се поделити на top-down и bottom-up у односу на почетну тачку предикције, почетна тачка може бити идентификовање особа или идентификовање делова тела.

Top-down методе прво предиктују особе на слици и генеришу позицију особе окружену правоугаоником, затим се врши препознавање позе за сваку особу понаособ. Супротно, bottom-up методе прво предиктују све делове тела сваке особе са улазне слике, а затим их групишу у односу на неки постојећи модел људског тела или неким другим алгоритмом.

Делови тела могу бити зглобови, удови или неки други мали делови тела. Са повећањем броја људи на слици, цена израчунавања за top-down методе се значајно повећава, док остаје иста за bottom-up методе, међутим, ако на слици постоје људи чији се делови тела преклапају, bottom-up методе се суочавају са изазовима да групишу одговарајуће делове тела код одређене особе.



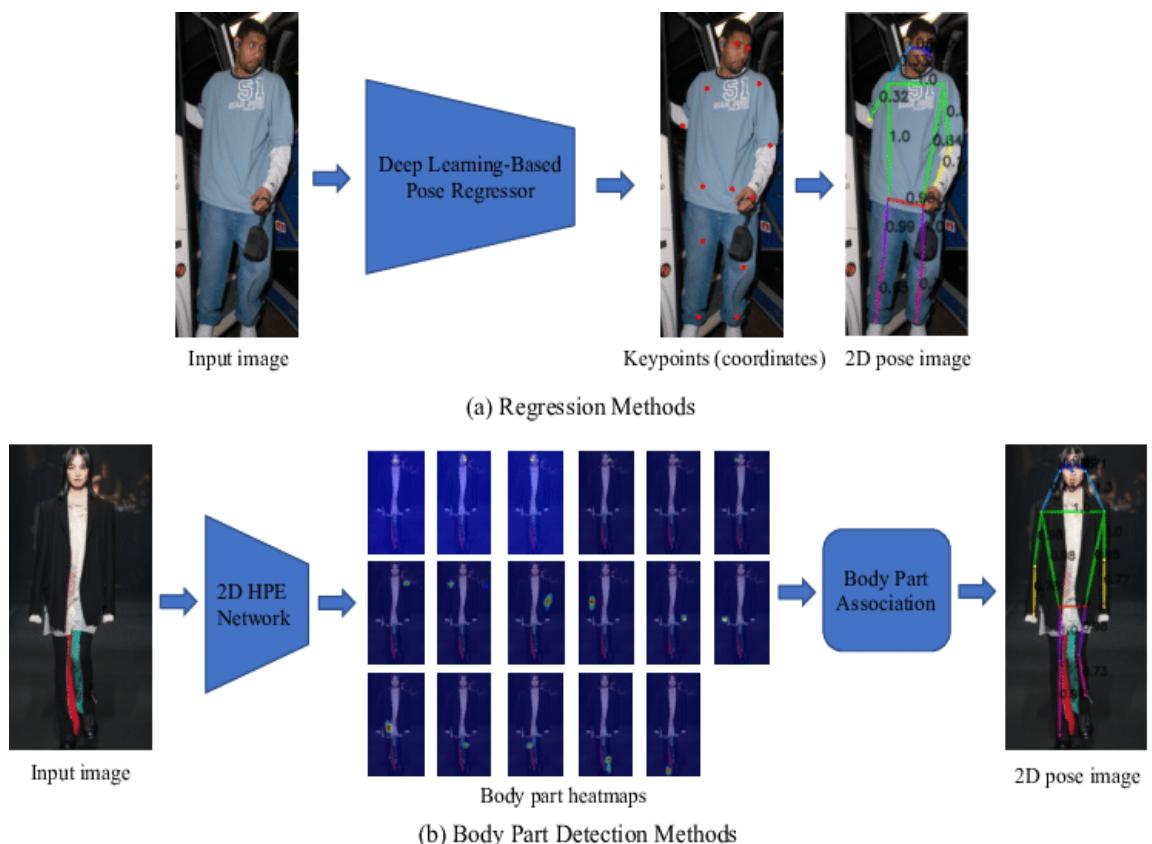
Слика 2.2. Приступи одозго на доле и одоздо на горе

### 2.2.3. Regression-based и Detection-based

На основу различитих формулатија проблема, методе процене људске позе засноване на дубоком учењу могу се поделити на методе засноване на регресији или на методе засноване на детекцији.

Методе засноване на регресији директно мапирају улазну слику у координате зглобова тела или параметре модела људског тела. Методе засноване на детекцији третирају делове тела као мете детекције на основу два широко коришћена приказа, а то су делови слика и топлотне мапе локација зглобова.

Директно мапирање зглобова са читавих слика је веома компликовано, јер је то нелинеаран проблем, док репрезентација малих региона омогућава јасније информације о пикселима и боље препознавање зглобова, али то даје одређени лимит тачности финалних координата зглобова.



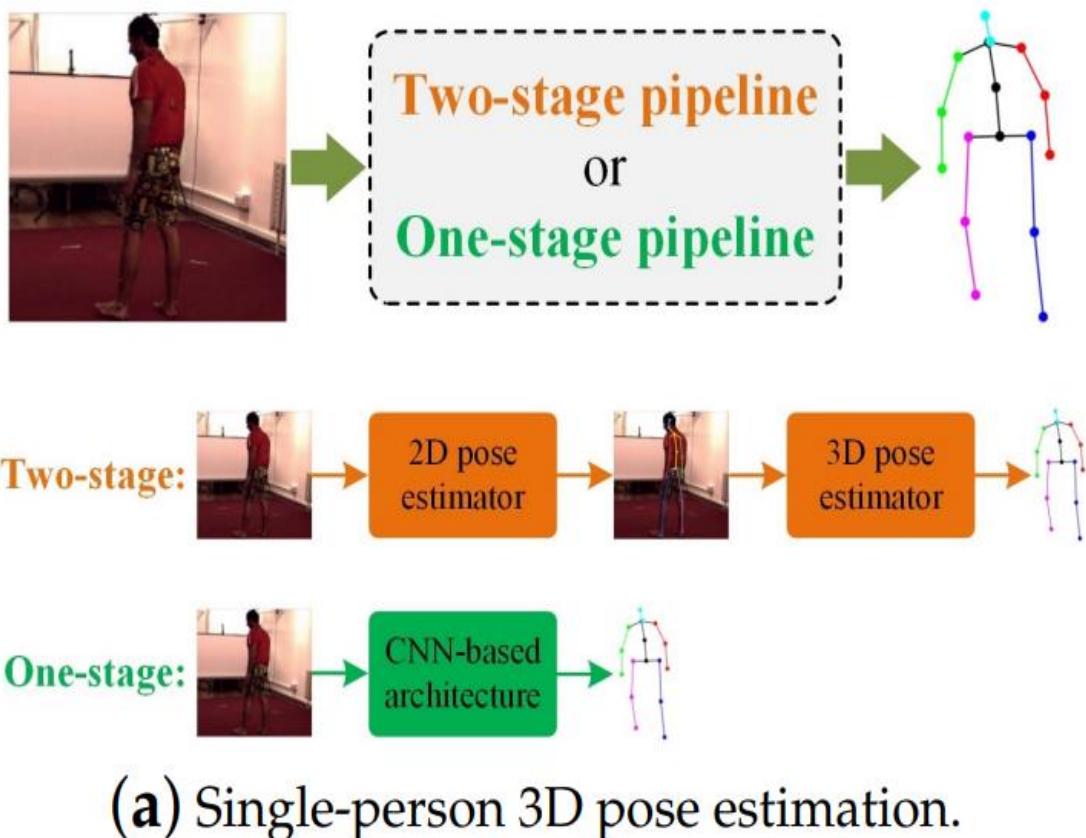
Слика 2.3. Приступ заснован на регресији и приступ заснован на топлотним мапама

#### 2.2.4. One-stage и Multi-stage

Једностепене методе засноване на дубоком учењу имају за циљ да мапирају улазну слику у људске позе коришћењем мреже од почетка до краја, док вишестепене методе предвиђају људску позу у више фаза.

На пример, неке методе за процену позе више особа прво откривају локације људи, а затим процењују људску позу за сваку отк rivenu особу. Друге 3D методе процене људске позе прво предвиђају локације зглобова на 2D површини, а затим их проширују на 3D простор.

Обука једностепених метода је лакша од вишеструких метода, али са мање ограничења и прецизности.



Слика 2.4. Једностепени и вишестепени приступ

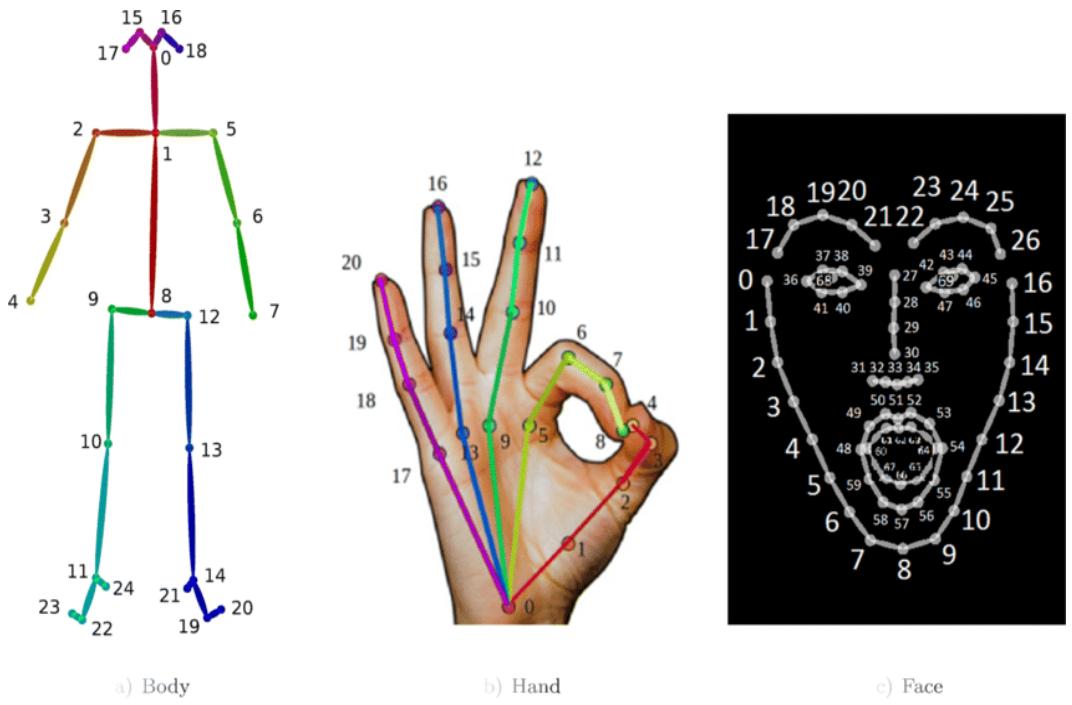
## **2.3. Библиотеке које омогућавају детекцију позе тела на основу слике**

### **2.3.1. OpenPose**

Постоји пуно библиотека за детекцију позе тела, свака од њих пружа специфичан приступ препознавању позе тела и има своје предности и мање у односу на остале библиотеке. Најпопуларнија библиотека овог типа јесте OpenPose,

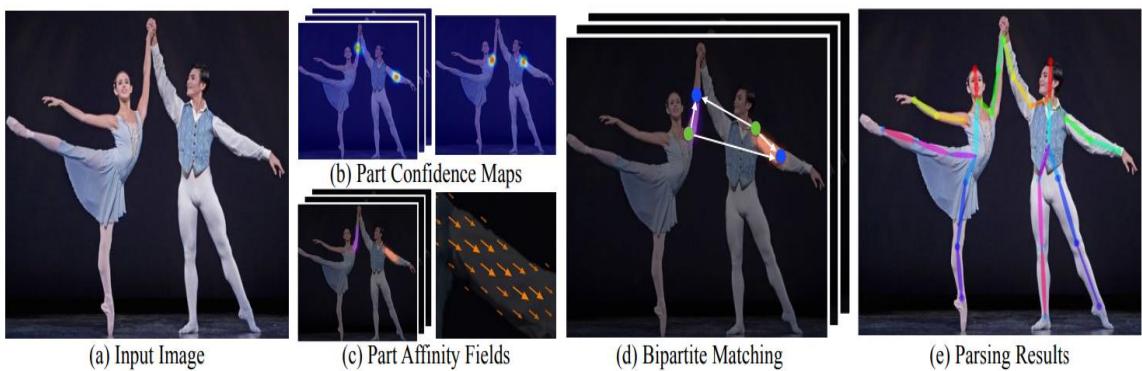
OpenPose може да препозна 137 кључних тачака на телу и његове функционалности су:

- 2D real-time препознавање кључних тачака једне или више особа на слици:
  - Процена 15, 18 или 25 кључних тачака за тело/стопало, укључујући и 6 тачака за стопало. Време препознавања кључних тачака у односу на број људи на слици је константно.
  - Процена 2x21 кључних тачака руке, за обе руке по 21. У овом случају време рада зависи од броја људи на слици.
  - Процена 70 кључних тачака лица. Време рада зависи од броја откривених људи на слици.
- 3D real-time препознавање више особа на слици:
  - 3D прорачун из више појединачних приказа
  - Урађена синхронизација са Flir (Infra-red) камерама
  - Компабилна технологија Flir/Point grey камерама
- Пакет алата за калибрацију камера:
  - Процена изобличења унутрашњих и спољашњих параметара камере
- Праћење особа



Слика 2.5. Кључне тачке на а) Телу, б) Руци и ц) Лицу

метод који користи је представник bottom-up приступа, што значи да се на слици прво детектују кључне тачке свих особа на слици са више или само са једном особом, а затим одређује којој особи припада која тачка. Последња верзија ове технологије је установљена 2019. године и додатак на претходни овакав рад[3] јесте нови приступ који користи непараметарску репрезентацију, која се назива поља афинитета делова енг. Part Affinity Fields - PAFs, како би научио да повеже делове тела са појединцима на слици. ПАФ представља 2D вектор поља који одређују локацију и оријентацију удова на слици. Овакав приступ је кључан за максимизацију тачности, узевши у обзир да побољшање детектованих локација није неопходно, тај део кода је избачен из CNN, док је самој конволутивној мрежи повећана дубина, овакав приступ је довео до повећања брзине за 200% и тачности за 7%, што је омогућило препознавање позе тела у времену веома близком реалном [5].



Слика 2.6. Процес препознавања позе тела код OpenPose

Једна од предности OpenPose јесте препознавање више особа на слици, на слици 2.6. може се видети целокупан процес обраде улазне фотографије. Као улаз користи се RGB фотографија величине  $w \times h$  (Слика 2.6. а.), а на излазу се добија фотографија са означеним кључним тачкама на телима свих особа са слике које су повезане у виду скелета. Прво се унапред предвиђа скуп сачињен од 2D мапа поверења (енг. Confidence maps) (Слика 2.6. б)  $S$  и скуп 2D векторских поља  $L$  сачињен од поља афинитета делова, који кодирају степен повезаности између делова тела (Слика 2.6. ц.)

Скуп  $S$  се састоји из укупно  $J$  мапа поверења, по једна за сваки део тела.

$$S = (S1, S2, \dots, SJ), \text{ где } Sj \in \mathbb{R}^{w \times h}, j \in \{1 \dots J\}.$$

Скуп  $L$  се састоји из укупно  $C$  вектора поља, по један вектор за сваки уд.

$$L = (L1, L2, \dots, LC), \text{ где } Lc \in \mathbb{R}^{w \times h \times 2}, c \in \{1 \dots C\}.$$

Парови делова тела су названи удови, како би биле јасније везе између делова, међутим неки делови тела нису људски удови (нпр. лице).

Конечно, мапе поверења  $S$  и поља афинитета делова  $L$  се прослеђују *greedy bipartite matching* алгоритму како би се добиле кључне тачке свих људи на слици.

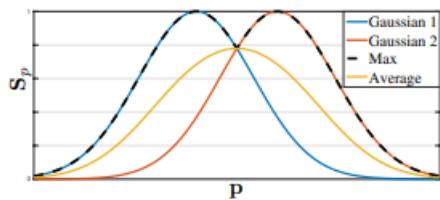
Свака мапа поверења је 2D приказ вероватноће да се одређени део тела може налазити на било ком датом пикселу. На пример, уколико се на слици налази само један човек, што је идеалан случај, очекује се да се у свакој мапи поверења за сваки део тела налази само један пик (максимална вероватноћа постојања дела тела на пикселу), сто ће водити ка томе да се на том месту налази одређени део тела. Међутим, у реалном случају са више особа на слици свака мапа поверења ће имати више пикова, односно по један пик за постојећи део тела на сваком телу са дате улазне слике, односно, један пик по видљивом делу тела  $j$  за сваку особу  $k$ . Мапе поверења за сваку особу  $k$  и сваки део тела  $j$  дефинисане су као:

$$S_{j,k}^*(p) = \exp \left( -\frac{\|p - x_{j,k}\|_2^2}{\sigma^2} \right)$$

то је Гаусова крива са постепеним променама где  $\sigma$  контролише ширење пика (максимума). Предвиђени пик из мреже је агрегација индивидуалних мапа поверења максималним оператором,

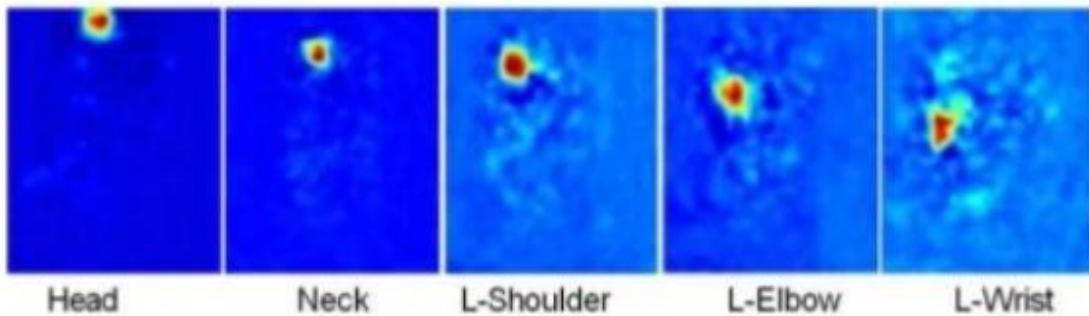
$$S_j^*(p) = \max_k S_{j,k}^*(p).$$

Узима се максимум мапа поверења уместо просека, тако да прецизност оближњих максимума остаје различита, што је илустровано на слици. У време тестирања се предвиђају мапе поверења и делови тела. На слици испод је приказан одабир максимума, где су испрекиданим линијама узете вредности које се користе за одређивање максимума.



*Слика 2.7. Одабир максимума*

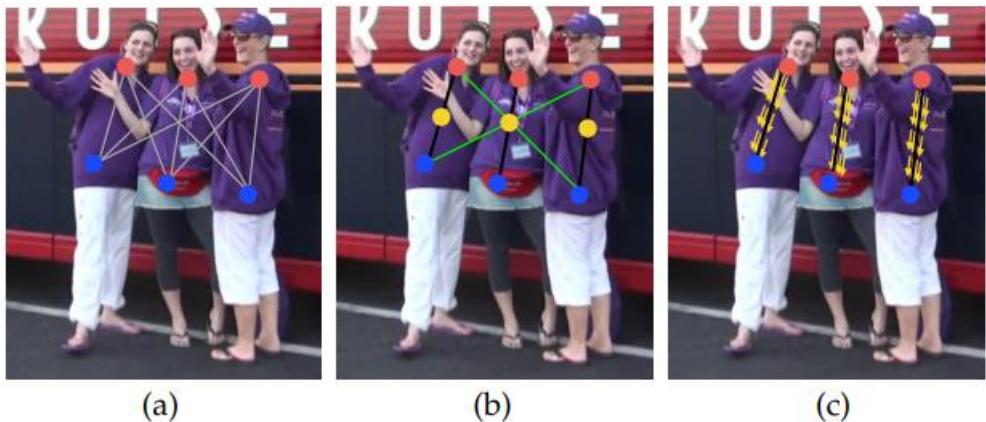
Ако неки део тела није препознат, уносе се координате  $x=0$  и  $y=0$ .



*Confidence map examples*

*Слика 2.8. Примери мапа поверења*

Како је задатак да се од тачака на телу и њихових веза креира скелетон за сваку особу на фотографији. Мапа поверења доприноси томе проналажењем делова тела на слици. Следећи проблем је повезивање делова тела на слици, како их саставити, а да се тачно зна који делови тела су које особе? Постоји више покушаја решења овог проблема, неки од њих су показани на слици испод



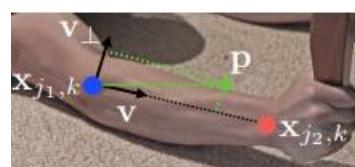
*Слика 2.9. Примери начина повезивања делова тела*

На слици 2.9. а., приказане су све могуће везе између делова тела особа на слици, ово нам не помаже да дођемо до правих веза. Покушај решавања јесте да се детектује додатна средња тачка између сваког пара делова на екстремитету и провери њена повезаност са тачкама кандидатима Слика 2.9. б., међутим, када се људи приближе једни другима, постојаће лажне повезаности и презентација повезаности неће бити исправна, проблем настаје због тога што се кодира положај, али не и оријентација екстремитета. Ова ограничења су решена уз помоћ поља афинитета дела тела. За сваки екстремитет постоји PAF који представља 2D вектор поља. За сваки пиксел у регији у којој се налази екстремитет, 2D вектор садржи смер те тачке од једне до друге стране екстремитета. Сваки тип екстремитета има PAF који представља спој између два одређена зглоба на једној особи.

На Слици 2.10., је приказан пример подлактице на којој  $x_{j1,k}$  и  $x_{j2,k}$  представљају позиције зглобова  $j1$  и  $j2$  екстремитета с особе  $k$  на слици. Уколико тачка  $p$  припада регији на којој се налази екстремитет вредност  $L_{c,k}^*(p)$  представља јединични вектор који је усмерен од  $j1$  према  $j2$ . За све остале тачке које се не налазе на регији, јединични вектор је 0.

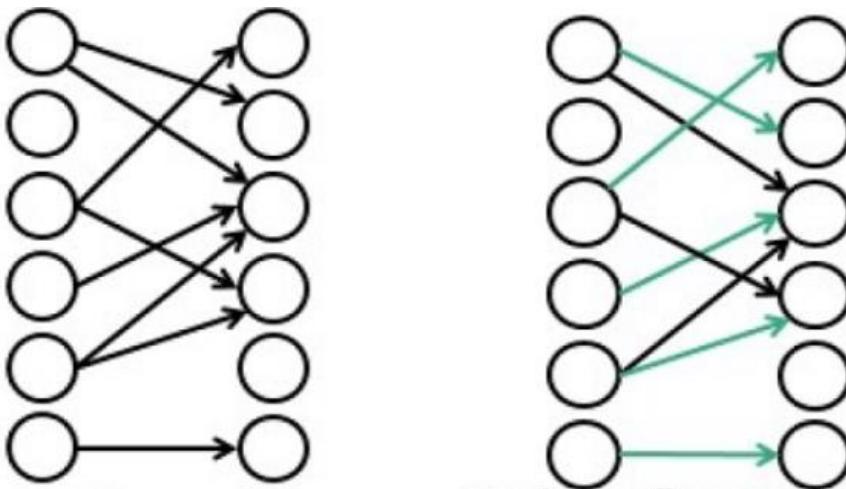
$$\mathbf{L}_{c,k}^*(\mathbf{p}) = \begin{cases} \mathbf{v}, & \text{ако је } p \text{ на екстремитету } c,k \\ \mathbf{0}, & \text{у осталим случајевима} \end{cases}$$

где је  $\mathbf{v} = (x_{j2,k} - x_{j1,k}) / \|x_{j2,k} - x_{j1,k}\|_2$  јединични вектор у правцу екстремитета.



*Слика 2.10. Одређивање правца повезаности зглобова*

И на крају, када имамо везе између зглобова, користимо *greedy bipartite matching* алгоритам који ради на принципу одабира делова тела који ће бити повезани за сваку особу, ако на слици имамо 5 особа, биће креирано 5 скелета, алгоритам у раду користи смерове екстремитета из PAF-а како би одредио повезаност свих делова тела, након овог корака, креирана је излазна слика. На слици 2.11. се налази пример графа који проналази везе између два дела тела на једној особи, овај алгоритам то ради за све тачке на телу и онда их међусобно повезује.



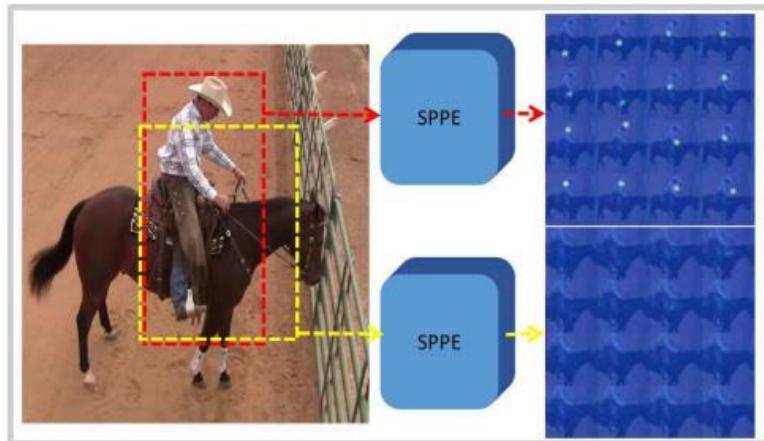
*Слика 2.11. Bipartite matching граф*

### 2.3.2. AlphaPose

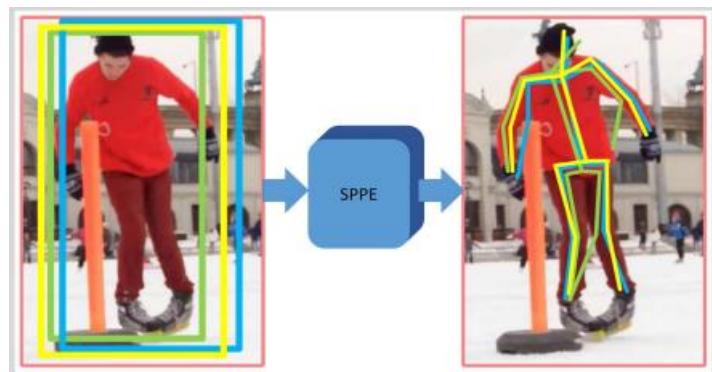
AlphaPose је још један представник библиотека које пружају могућност препознавања позе тела за препознавање више особа на слици у реалном времену, препознаје 136 тачака на телу човека, укључујући тело, руке и главу. Метод који користи јесте регионална процена позе више особа (енг. Regional Multi-Person Pose Estimation – RMPE). Овај метод користи top-down приступ, што значи да се прво детектују особе на слици, а затим њихови делови тела и везе између делова тела.

Шта је проблем са оваквим приступом? Једноставно, при детекцији особа на улазној фотографији, у старијим алгоритмима је долазило до грешке тако да је једна особа препозната више пута на слици и да је око ње била описана такозвана „оквирнија кутија“ (енг. rounding box) која не прекрива цело тело, што је доводило до тога да се неки делови тела не препознају, а истовремено се алгоритам беспотребно извршавао над већим бројем кутија за једну особу, што је наравно утицало на брзину и исправност препознавања позе тела.

Овај метод је креиран како би се лакше препознале позе тела у присуству више оквирних кутија Слика 2.13. и решио проблем грешке локализације тела (Слика 2.12.).



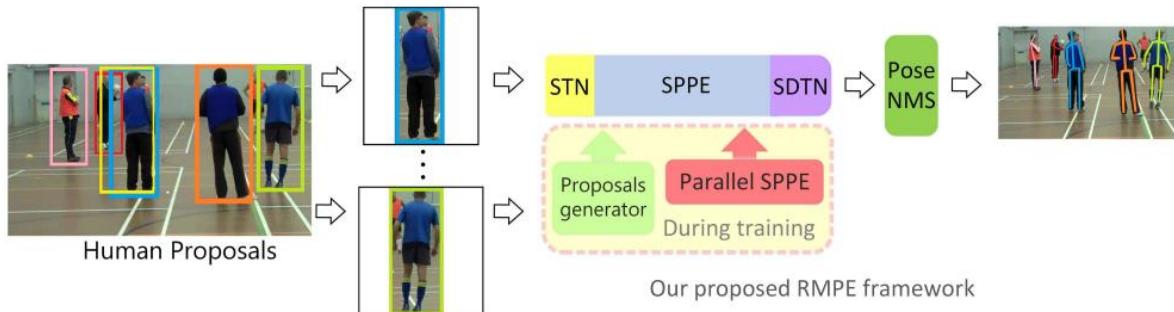
*Слика 2.12. Пример грешке локализације тела*



*Слика 2.13. Проблем више оквирних кутија*

Овај метод побољшава перформансе алгоритама за процену људске позе заснованих на SPPE (енг. Single Person Pose Estimation). Дизајнирана је нова просторна трансформаторска мрежа SSTN (енг. Symetric Spatial Transformer Network) која је приклучена на SPPE да би издвојила висококвалитетни регион једне особе из нетачне граничне кутије. Такође је уведена нова паралелна SPPE грана како би се та мрежа оптимизовала [6].

Да би се решио проблем сувишне детекције, уведено је не-максимално потискивање NMS (енг. non-maximum suppression). Ова параметарска поза елиминише сувишне позе користећи нову метрику удаљености поза за упоређивање сличности позе. Приступ вођен подацима се користи да оптимизује параметре удаљености позе. На крају додат је нови генератор предлога људских поза PGPG (енг. pose-guided human proposal generator), за повећање узорака обуке. Учењем са излазних фотографија које препознају различите позе може се симулирати генерирање граничних кутија чиме се производи велика количина података за обуку и бољи рад алгоритма.



Слика 2.14. Ток препознавања више особа на слици уз помоћ RMPE

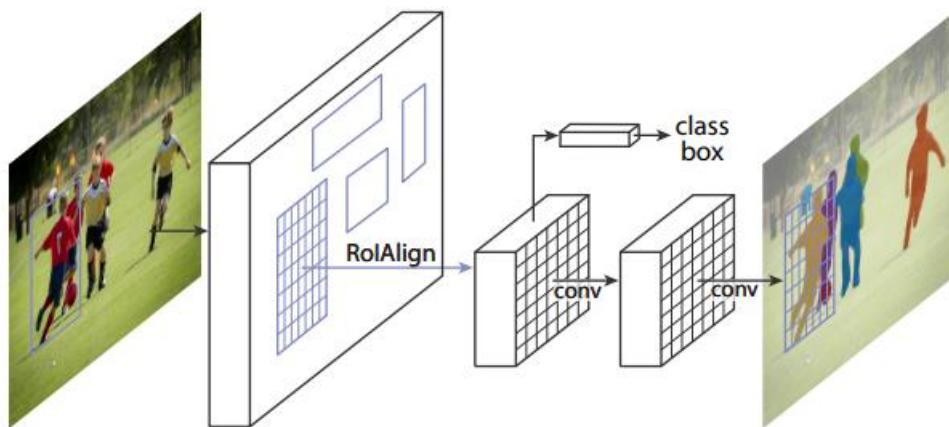
На слици изнад (Слика 2.14.) је представљен ток препознавања поза више тела на слици уз помоћ RMPE модела. На улазној слици се препознају особе, међутим, предлози особа који се добијају нису прилагођени SPPE, доказано је да се мали превод или изрезивање људских предлога може значајно добро одразити на перформансе SPPE, због тога се уводи STN као и паралелни SPPE како би се побољшали услови за SPPE када су дати несавршени предлози особа на слици. STN издава висококвалитетне регионе доминантних људских предлога уз помоћ паралелног SPPE, разлика између SPPE и паралелног SPPE јесте у томе што су неке фазе извршења изостављене и паралелни SPPE је од велике помоћи STN-у при фокусирању на исправне области и издавању висококвалитетних региона, он се користи само у фази тренинга. SDTN служи да генерише предлоге поза особа, док NMS служи како би се слиминисале сувишне позе које су добијене за исте особе и изабере најтачнију, прво се бира најпоузданјија поза као референтна, а позе које су јој блиске подлежу елиминацији, овај процес се понавља на свим позама док се све сувишне позе не елиминишу и док се не појаве само јединствене позе. Елиминација се врши тако што се дефинише метрика дистанце позе  $d(P_i, P_j | \Lambda)$  за мерење сличности позе и праг  $\eta$  као елиминацијони критеријум, где је  $\Lambda$  скуп параметара функције  $d(\cdot)$ :

$$f(P_i, P_j | \Lambda, \eta) = \mathbb{1}[d(P_i, P_j | \Lambda, \lambda) \leq \eta]$$

Ако је  $d(\cdot)$  мање од  $\eta$  излаз функције  $f$  треба да буде 1, што говори да поза  $P_i$  треба да буде елиминисана због сличности са позом  $P_j$ . Након овог корака добија се излазна слика без понављања скелета и сувишних оквирних кутија.

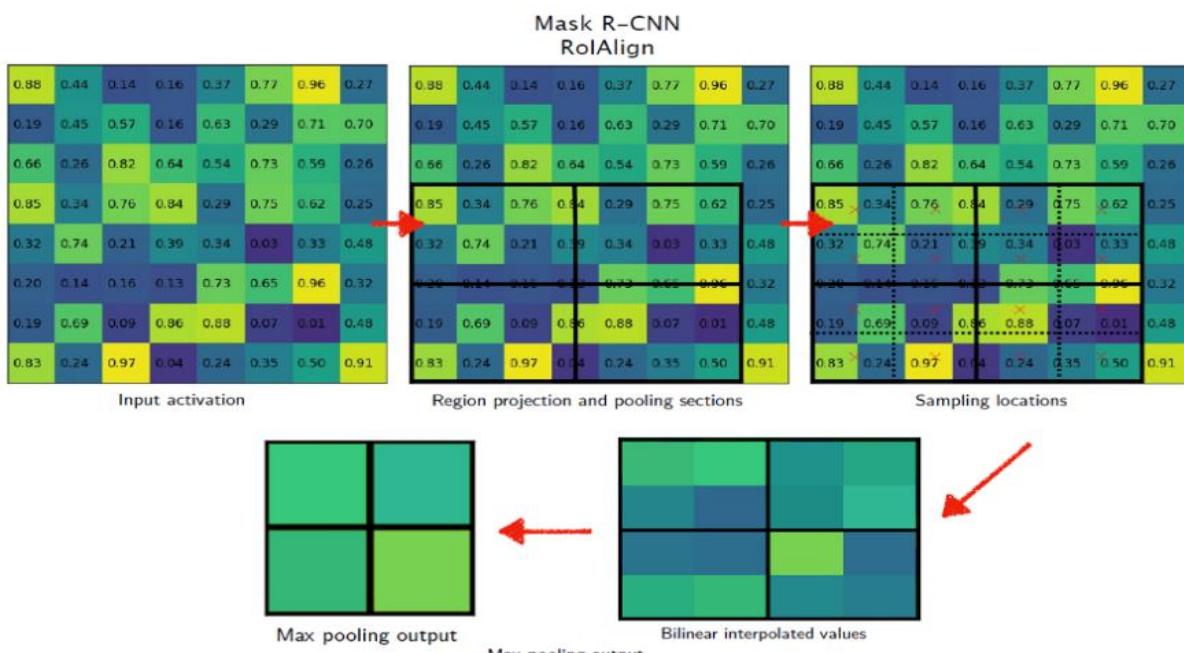
### 2.3.3. Detectron

Mask R-CNN [7] или Detectron, препознаје 17 кључних тачака на телу човека по COCO стандарду, представља модел који је заснован искључиво на дубоком учењу, омогућава детекцију објекта на слици док истовремено генерише висококвалитетну маску сегментације за сваку инстанцу објекта, ова метода проширује Faster R-CNN метод, додавањем гране за предвиђање маске објекта паралелно са постојећом граном за препознавање граничних оквира, овај метод је лако генерализовати за друге задатке, као што је препознавање позе човека у већ пронађеној оквирној кутији.



Слика 2.15. Архитектура Mask R-CNN алгоритма

Предвиђање маски сегментације се врши на сваком региону од интереса RoI (енг. Region of Interest). Грана маске је мала потпуна конволуциона мрежа FCN (енг. Fully Convolutional Network), примењена на сваки регион од интереса, предвиђајући маску сегментације у pixel-to-pixel маниру. Mask R-CNN је једноставан за имплементацију, с обзиром на Faster R-CNN оквир који омогућава широк спектар флексибилних дизајна архитектуре. Поред тога, додатна грана за рачунање маске додаје веома мале рачунске трошкове постојећем методу, тако да је могуће брзо експериментисање. Правилна конструкција гране маске је кључна за добре резултате. Пошто Faster R-CNN није дизајниран за pixel-to-pixel поравнање између улаза и излаза, па због тога RoIPool, дефакто основна операција за праћење инстанци, изводи грубу просторну квантацију за екстракцију карактеристика. Да би се поправила неусклађеност, предложен је слој без квантације, RoIAlign који чува тачне просторне локације, овај додатак побољшава тачност маске за 10-50% показујући веће добитке за строже праћење локација.



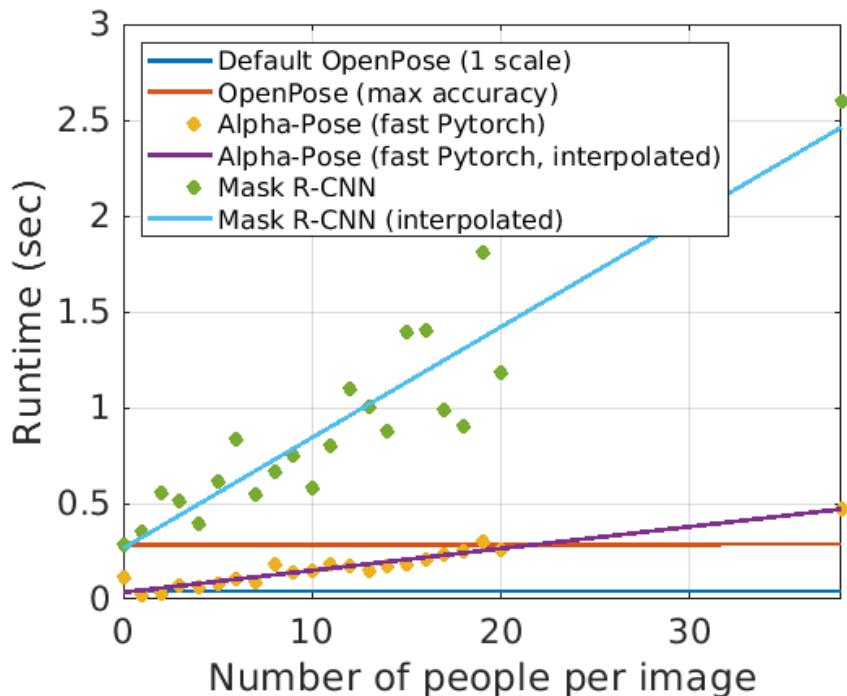
Слика 2.16. Mask R-CNN RoIAlign

На слици 2.16. је приказан пример RoIAlignPool операције, за разлику од *Faster R-CNN* где се за целобројну дужину узимају правоугаоници различите прилагођене величине, код Mask R-CNN-а се за дужину узимају правоугаоници једнаке величине. На основу преклапања области са вредностима мапе обележја, користи се би-линеарна интерполација да би се добила средња обједињена мапа обележја, након тога се врши максимално обједињавање над овом средње груписаном мапом обележја.

Такође одвојена је маска од предвиђања класе, предвиђа се бинарна маска за сваку класу независно, без конкуренције међу класама, и ослања се на грани класификације RoI мреже да би се предвидела категорија. Насупрот томе FCN обавља вишекласну категоризацију по пикселу, која спаја сегментацију и класификацију, што на основу изведенних експеримената функционише лоше.

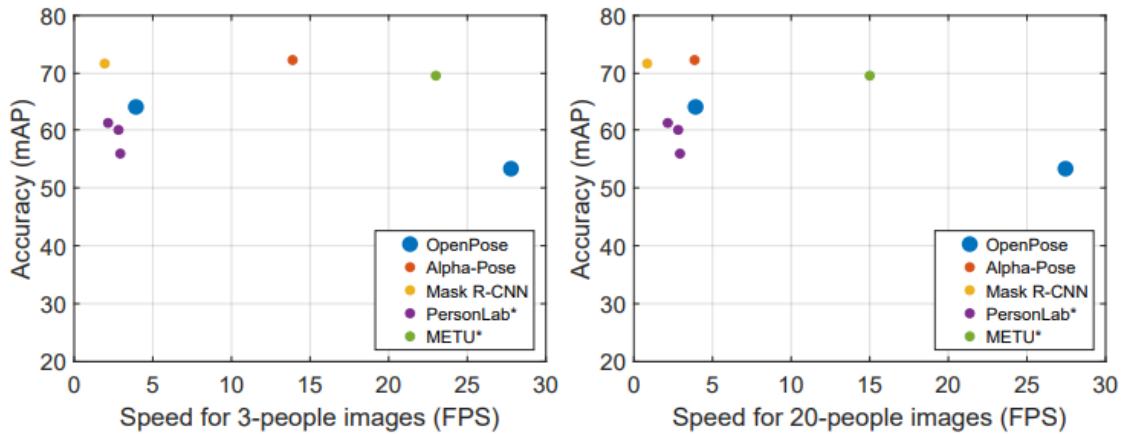
Овај метод је креиран како би био прилагодљив разним ситним изменама које могу довести до запаљујућих резултата, па је тако лако проширљив и за додавање опције препознавања људске позе. Ради тако што се кључне тачке на телу особе препознају као “врући делови” тачније за сваку кључну тачку на телу се креира маска где је јасно обележен део тела за који је та маска задужена. Затим се ти делови спајају и добија се потпуна слика скелета.

## 2.4. Поређење библиотека за детекцију позе тела



Слика 2.17. Поређење времена извршења у односу на број људи по слици

На слици изнад извршено је поређење времена закључивања између OpenPose, Mask R-CNN и AlphaPose алгоритама, тестирање је изведено на систему са графичком картицом NVIDIA 1080 и са технологијом CUDA8. Анализа је извршена коришћењем истих фотографија за сваки алгоритам и поновљена 1000 пута, а затим је израчуната средња вредност резултата. Са слике се види да је време које је потребно OpenPose алгоритму за рачунање поза људи са слике непроменљиво у односу на број људи на слици, док је с друге стране код остала два алгоритма у линеарној зависности од броја људи и расте са повећањем броја људи на слици.



Слика 2.18. Поређење брзине и тачности процењене

Са графика на слици 2.18. се може закључити да методе које користе top-down приступ (AplhaPose, Mask R-CNN) доприносе већој тачности, али је брзина извршавања процене мања него код bottom-up (OpenPose, PersonLab, METU) приступа. Главни разлог за мању тачност bottom-up приступа јесте њихова ограничена резолуција, док top-down методе појединачно исецају сваку особу и уносе податке у своје мреже, супротан приступ мора да обрађује читаву слику, са напретком технологије ове методе смањују јаз у тачности приказа. Ако се тражи највећа тачност, био би одабран AlphaPose, док се за највећу брзину узима OpenPose, они су такође по коришћењу најпопуларнији методи својих приступа. Top-down приступи омогућавају боље резултате за слике са неколико људи, али брзина експоненцијално опада са повећањем броја људи

Method	CUDA	CPU-only
Original MPII model	73 ms	2309 ms
Original COCO model	74 ms	2407 ms
Body+foot model	36 ms	10396 ms

Слика 2.19. Разлика у времену препознавања

На слици 2.19. је приказана анализа разлике у времену препознавања између почетних MPII модела, COCO модела и новог модела који представља комбинацију препознавања тела и стопала. Занимљиво је да је нови модел два пута бржи када се

користи GPU верзија, док је време рада за CPU верзију чак пет пута спорије у поређењу са оригиналним MPII моделом. Нова архитектура захтева много више меморије, а мање је операција које се требају извршити, графичке картице имају више користи од смањења броја операција, док је CPU верзија знатно спорија због већих захтева за меморијом. Перформансе OpenGL-а и CUDA-е се не могу директно поредити, јер захтевају различит хардвер, посебно различите марке GPU-а, што је велики проблем са CUDA-ом.

*Табела 1. Поређење различитих библиотека по броју кључних тачака*

Детектор	Тело	Руке	Лице	Праћење људи
OpenPose	25	42	70	jedan
	-	-	-	vise
Detectron	17	-	-	vise
	-	-	-	-
AlphaPose	26	42	68	vise
	-	-	-	-
MediaPipe	33	42	468	vise

## 2.5. MediaPipe

MediaPipe представља open-source framework који пружа прилагодљива решења заснована на машинском учењу за дељење и презентовање видео снимака у реалном времену (eng. real-time).

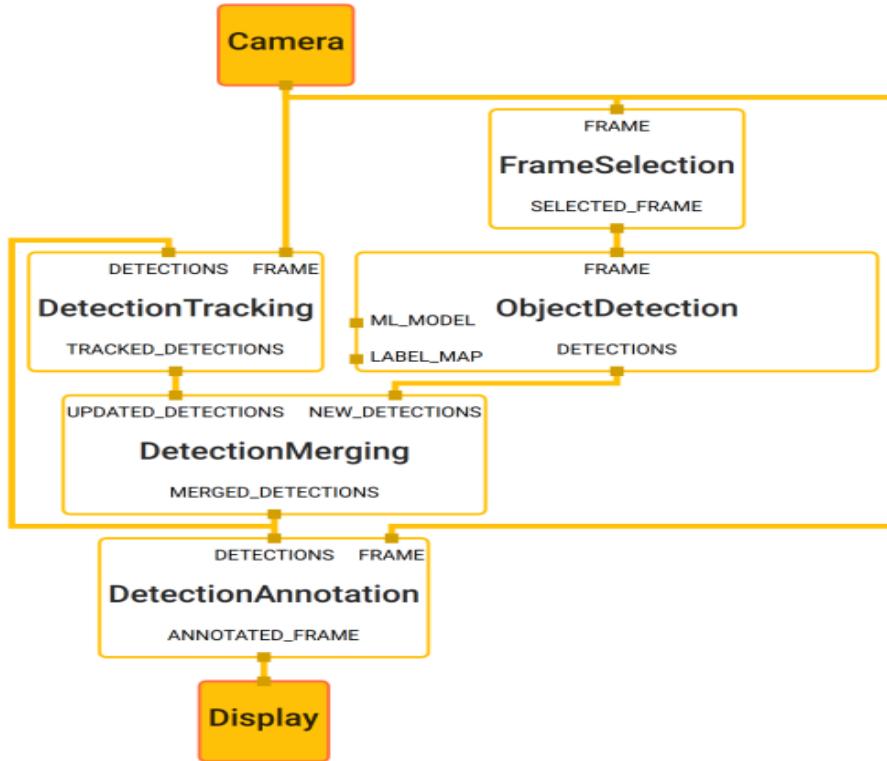
Дизајниран је за истраживаче, студенте и прогамере који имплементирају апликације засноване на машинском учењу, објављују код који прати истраживачки рад и граде прототипове технологије. Главни случај употребе ове технологије јесте брзо прављење прототипа апликација које раде на основу паметног закључивања и компоненти које се могу користити више пута. Омогућава инкрементални напредак компоненти и апликација.

Модификовање апликације за паметно учење да би се додали додатни кораци обраде или модели закључивања може бити веома тешко због превелике повезаности међу корацима, поред тога, развој исте апликације за различите платформе одузима много времена и обично укључује оптимизацију корака закључивања и обраде, да би се исправно и ефикасно покренули на циљном уређају

MediaPipe решава ове изазове апстрактовањем и повезивањем појединачних модела закључивања у систем проточне обраде за одржавање. Сви кораци неопходни за закључивање из сензорских података и да се добију уочени резултати наведени су у конфигурацији проточне обраде. Лако је поново користити MediaPipe компоненте у различитим проточним обрадама у узастопним апликацијама, јер ове компоненте деле заједнички интерфејс оријентисан на податке из временских серија података, Свака линија проточне обраде тада може да ради са истим понашањем на различитим платформама, омогућавајући ономе ко развија апликацију, да је развије на радним станицама (компјутерима), а да је, на пример, примени на мобилном уређају.

MediaPipe се састоји од три основне компоненте:

- Оквир за закључивање на основу сензорских података
- Скуп алата за процену перформанси
- Колекције компоненти за закључивање и обраду података које се могу поново користити, које се називају калкулатори



*Слика 2.20. Архитектура код MediaPipe [8]*

MediaPipe дозвољава девелоперу да постепено надограђује скуп елемената за обраду података. Скуп се дефинише као усмерени граф компоненти где је свака компонента калкулатор. Граф се креира уз помоћ GraphConfig протокол бафера а затим покреће коришћењем Graph Object-а. У графу, калкулатори су повезани токовима података (енг. Data Streams). Сваки ток представља временску серију пакета података. Заједно калкулатори и токови дефинишу data-flow граф. Пакети који иду кроз граф су разврстани према својим временским ознакама унутар временске серије. Скуп се може надоградити уметањем или заменом калкулатора било где у графикону. Улазни подаци морају бити истог типа као што су излазни подаци калкулатора који их обрађује. Програмери могу сами дефинисати одређене калкулаторе. Док граф паралелно извршава калкулаторе, сваки калкулатор извршава највише једну нит. Ово ограничење, уз непроменљивост пакета података обезбеђује да се калкулатори могу дефинисати без специјализоване експертизе у вешенитном програмирању.

Ова технологија се користи у рачунарском виду и веома је корисна за многе области које рачунарски вид обухвата. Примењива је на различитим платформама, у различитим програмским језицима и има разне области примене. Развијена је пре свега за детекцију и препознавање покрета руку, међутим проширила се на многе области као што су: препознавање лица, препознавање мреже лица, препознавање позе, сегментацију косе, селфи сегментацију, детекцију објекта, праћење објекта кутија и многе друге.

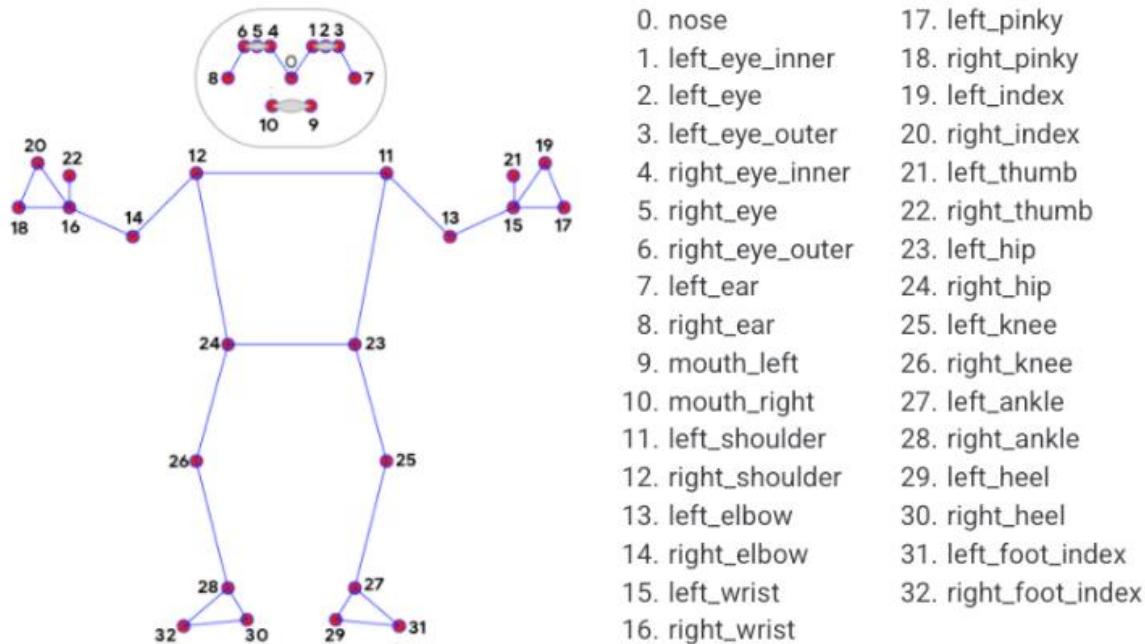
	Android	iOS	C++	Python	JS
Face Detection	✓	✓	✓	✓	✓
Face Mesh	✓	✓	✓	✓	✓
Iris	✓	✓	✓		
Hands	✓	✓	✓	✓	✓
Pose	✓	✓	✓	✓	✓
Holistic	✓	✓	✓	✓	✓
Selfie Segmentation	✓	✓	✓	✓	✓
Hair Segmentation	✓		✓		
Object Detection	✓	✓	✓		
Box Tracking	✓	✓	✓		
Instant Motion Tracking	✓				
Objectron	✓		✓	✓	✓
KNIFT	✓				
AutoFlip			✓		
MediaSequence			✓		
YouTube 8M			✓		

Слика 2.21. Примена MediaPipe-а

Поред наведених програмских језика [9], MediaPipe библиотека се може користити и у Coral програмском језику за детекцију лица и детекцију објекта, што је још један доказ да полако заузима водећу позицију у препознавању и обради видео снимака и фотографија. Док остале библиотеке углавном омогућавају рад на моћним десктоп рачунарима, MediaPipe се може користити на најmodернијим паметним телефонима, рачунарима или лаптоповима.

Што се тиче препознавања позе тела човека, MediaPipe пружа до сада највише кључних тачака у препознавању, чак 33 и то у виду 3D кључних тачака или како се у овом framework-у називају обележја (енг. **Landmarks**), у наставку ће бити коришћен израз кључне тачке како би било јасно поређење са осталим технологијама, такође, могуће је креирати маску сегментације позадине на целом телу из RGB видео записа користећи **BlazePose** метод за препознавање позе тела. Кључне тачке подразумевају додатак у виду препознавања кључних тачака шаке и препознавања кључних тачака стопала, што је од круцијалне важности за демо апликацију која ће бити презентована у

наставку.



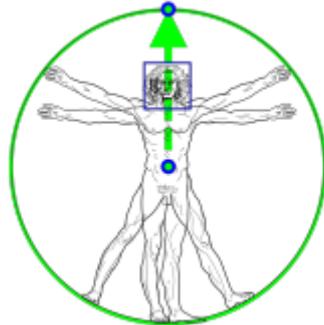
Слика 2.22. Кључне тачке на телу - MediaPipe

Као што је раније речено, метод који користи MediaPipe за детекцију позе тела човека јесте BlazePose, који се базира на регресији (енг. Regression-based). BlazePose је „lightweight“ CNN за детекцију позе човека која је имплементирана за real-time рад на мобилним уређајима. Ово га чини погодним за праћење фитнес покрета и препознавање знаковног језика. Извршена је надоградња за препознавање позе тела човека која користи топлотне мапе и регресију истовремено.

Уобичајени приступ је да се произведу топлотне мапе за сваки део тела са смањењем одступања за сваку координату, иако се овај модел лако прилагођава раду са више особа на слици, он чини модел за препознавање једне особе много комплекснијим него што је погодан за закључивање у реалном времену. За разлику од техника заснованих на топлотној мапи, приступи засновани на регресији, иако су мање захтевни са стране хардвера али и скалабилнији, покушавају да предвиде средње вредности координата, често наилазе на проблем двосмислености због особа које заклањају једна другу, овај проблем је захтевно решити. За решење овог проблема користи се енкодер-декодер мрежна архитектура за предвиђање топлотних мапа свих кључних тачака праћена још једним енкодером израчунава тачне координате кључних тачака. Закључак је да се грана топлотне мапе може избацити током закључивања, што алгоритам чини довољно лаганим за покретање на мобилном телефону.

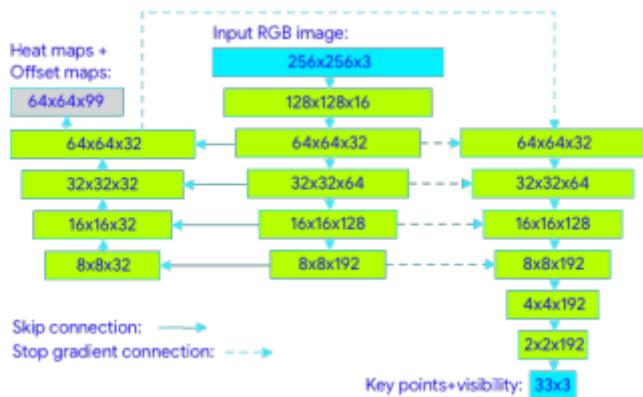
Савремена решења се ослањају на NMS алгоритам за последњи корак обраде препознавања објекта. Алгоритам добро функционише за статичне објекте који не могу много да промене свој облик, међутим, људско тело може да се нађе у разноразним положајима, да би се ово превазишло креатори су се фокусирали на откривање оквирних кутија релативно крутих делова тела код човека, као што су људско лице или торзо.

Приметили су да је у многим случајевима најјачи сигнал неуронској мрежи о положају торза само лице особе. Како би детектор позе био учињен лаганијим и бржим, узето је у обзир да за претпоставку глава особе увек треба да буде видљива како би се проценила поза тела.



*Слика 2.23. Поређење BlazePose детектора и детектора лица.*

Као последица тога, креира се нова врста детектора оквирне кутије као основа за детекцију особе Слика 2.23. Овај детектор осим лица предвиђа додатне параметре поравнања специфичне за особу: средњу тачку између бокова особе, величину круга који окружује целу особу и нагиб који представља угао између линија које повезују две тачке средњег рамена и средине кукова. На слици 2.23. приказано је поређење детектора који се користи у односу на класични детектор лица, поравњава се особа тако да се тачка између кукова налази у центру квадратне слике која се преноси као улаз неуронске мреже, процењује се ротација као линија L између тачака средњег бока и средине рамена и ротира слика тако да је L паралелна са у-осом. Скала је процењена тако да се све тачке тела уклапају у квадратну кутију описану око тела. BlazePose такође омогућава и препознавање позе тела и када комплетна особа није на слици, уводи се класификатор видљивости (енг. Visibility), који поред три координате за сваку тачку говори о томе да ли је одређена тачка заклоњена и да ли се њен положај сматра нетачним.



*Слика 2.24. Мрежсна архитектура BlazePose*

На слици 2.24. приказана је мрежна архитектура метода, присвојена је комбинација топлотних мапа, офсета и приступа заснованог на регресији, где се топлотне мапе и офсет користе само у фази тренинга, а затим резултате тога користи регресиони кодер како би добили коначан облик резултата а то је 33 кључних тачака од којих свака има 3 координате у простору.

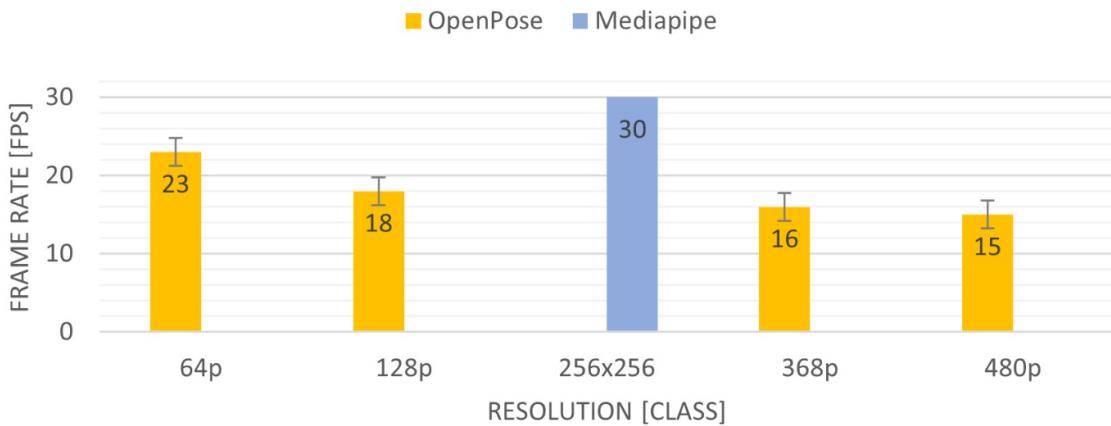
Овај приступ је инспирисан „Stacked Hourglass [10]“ приступом, али се у овом приступу користи мрежа декодер-енкодер, користи се прескакање између свих фаза мреже да би се постигла равнотежа између функција високог и ниског нивоа, док се градијенти из регресионог енкодера не преносе назад на карактеристике које су обучене за топлотну мапу. Откривено је да ово не само да побољшава предвиђања топлотне карте, већ и значајно повећава тачност регресије координата.

## 2.6. Зашто MediaPipe?

При одлучивању библиотеке која ће бити коришћена за пројекат препознавања сигнала фудбалског судије помоћника установљено је да ће пројекат бити демо типа и да његова имплементација захтева технологију која пружа тачне координате зглобова у 2D или 3D простору, координате кључних тачака шаке тј. палац, мали прст и кажипрст, праћење покрета једне особе у реалном времену са видео записа, рачунање координата када оне буду прекривене неким другим делом тела, технологију која не зависи много од замућења слике и лошег осветљења, која омогућава класификацију позе тела, ради са што већим бројем фрејмова по секунди, има велику тачност и нема ограничења платформе или уређаја на коме се може користити.

Истраживањем је закључено да су најпопуларније библиотеке које делимично задовољавају услове за креирање тражене апликације MediaPipe и OpenPose, обе технологије задовољавају потребе за кључним тачкама на телу човека које су захтеване, са малом разликом, где би за препознавање круцијалних прстију на руци, код OpenPose морала да се позивају два метода, а то су метод за детекцију шаке и метод за детекцију позе тела, док MediaPipe даје могућност да извршавањем само једног алгоритма добијемо тражене кључне тацке, затим, како би се добиле 3D координате код OpenPose је потребно имати више камера различитог типа и обраду података само на рачунару, што би захтевало више хардверских ресурса, али и представљало већи проблем за обраду података са више различитих истовремених снимака, док супарничка технологија све то омогућава једноставним снимком са обичне RGB камере.

## Frame Rate by Resolution

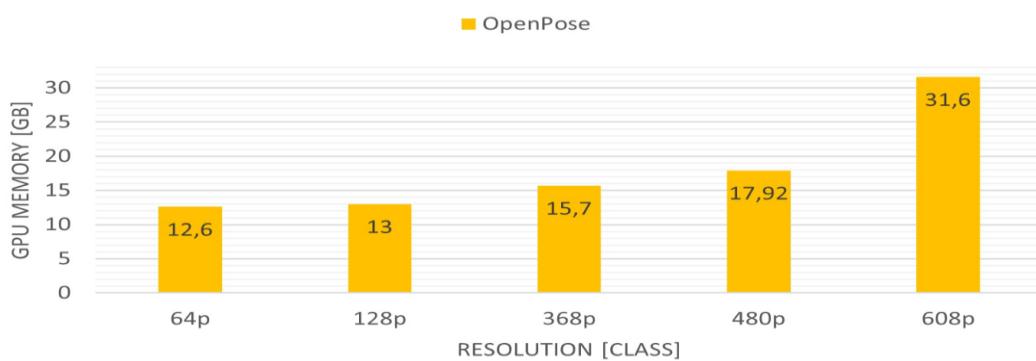


Слика 2.25. Број frame-ова у односу на резолуцију

Кад је у питању брзина и подржавање резолуције, на основу поређења [11], које је извршено на рачунару са Linux оперативним системом, са процесором Intel (R) Xeon (R) Platinum 8270 CPU @ 2.70GHz – 26 cores, затим графичком картицом NVIDIA GeForce RTX 2080 Ti са 11 GB RAM-а x2 у програмском језику Python, на слици 2.25. су приказани резултати поређења.

Што показује да са повећањем резолуције слике (слика 2.25.) OpenPose ради све спорије, а како се смањивањем резолуције слике смањује и тачност одређених координата, ова технологија није погодна за висококвалитетне фотографије, док с друге стране MediaPipe са својим основним подешавањем и обрадом 30FPS (енг. Frames Per Second) даје сасвим задовољавајуће резултате, још један проблем који настаје јесте да са повећањем резолуције потражња за меморијом GPU постаје све већа слика 2.26.

## GPU Memory Usage by Resolution



Слика 2.26. Повећање захтева за меморијом са повећањем резолуције

Исто истраживање[8], са датим перформансама рачунара, је показало да је MediaPipe дао за нијансу боље резултате од супарничке технологије када је у питању осветљење слике. Такође је доказано да када се покривају делови тела од стране других делова тела MediaPipe даје боље резултате када се одређују кључне тачке на шаци, док OpenPose даје боље резултате када се ради о телу, што је још један показатељ, обзиром на захтеве демо апликације, а то је да може доћи баш до преклапања кључних тачака на руци, да је MediaPipe боље решење.

Такође, при коришћењу OpenPose библиотеке, за брзину и тачност резултата морала би се инсталирати CUDA технологија, која је хардверски захтевна и не подржава све GPU, што доводи до закључка да је могућност извршења овакве апликације са том технологијом на различитим уређајима веома упитна и да ће само одређени рачунари бити у могућности да обрађују захтеване податке, што девелоперима прави велике проблеме у самом старту.

Креатори MediaPipe-а су своју библиотеку иницијално наменили апликацијама за препознавање шаке и покрета руке и апликацијама које се заснивају на процени положаја фитнес и спортских покрета, што води ка томе да је ова библиотека прави избор за тип демо апликације која се жели имплементирати.

Закључак, када је у питању апликација спортског типа где се проверава и процењује исправност покрета и урађених вежби/сигнала, коју ће користити спортисти на терену или у сали, MediaPipe представља најбољи избор библиотеке за препознавање позе тела.

### **3. Препознавање сигнала фудбалског судије помоћника**

Фудбалски судија помоћник има тежак задатак, током фудбалских мечева, он треба својим телом, без иједне речи да свим присутним људима на терену и ван њега да до знања какав је прекршај фудбалске игре сигнализирао. Сигнали су разноврсни и могу се састојати из једног или два комбинована корака, гледано са становишта рачунарског вида.

Сигнали који се сastoјe из јedнog коракa служe какo бi за вeћ jасно прекинуту игру (лопта изашла ван терена) одредили којa ћe екипа наставити игру и на који начин (убацивањe, ударац из угla, ударац сa вратa).

Сигнали који се сastoјe из два комбинована коракa су комплексниji, први корак може бити у две варијантe, прва варијанта је када судија помоћник жели да да сигнализира сe игра треба прекинути (офсајд, лопта није jасно напустила терен), а затим након сигнализацијe прекида показујe какo игру треба наставити, у овом случајu, судија помоћник прво подиже руку усправно, задржава јe горе неколико секунди (док главни судија прекине игру), након тога показујe адекватан наставак игре.

Друга варијанта јесте сигнализирањe прекршајa, прекршај сe сигнализира тако што сe јedна од две руке подигне усправно, а затим сe њом “маше” какo бi главни судија прекину игру, након тога сe показујe да ли јe прекршај за одбрамбену или нападајuћu страну.

Кao што јe у претходном тексту наведено, из угla препознавањe сигнала судије помоћника уз помоћ рачунарског вида, сигнале делимо на сигнале из јedнog и два коракa:

Сигнали који сe сastoјe из јedнog коракa:

- Убацивањe за нападајuћu страну
- Убацивањe за одбрамбену страну
- Ударац сa вратa
- Ударац из угla
- Замена

- Прекид



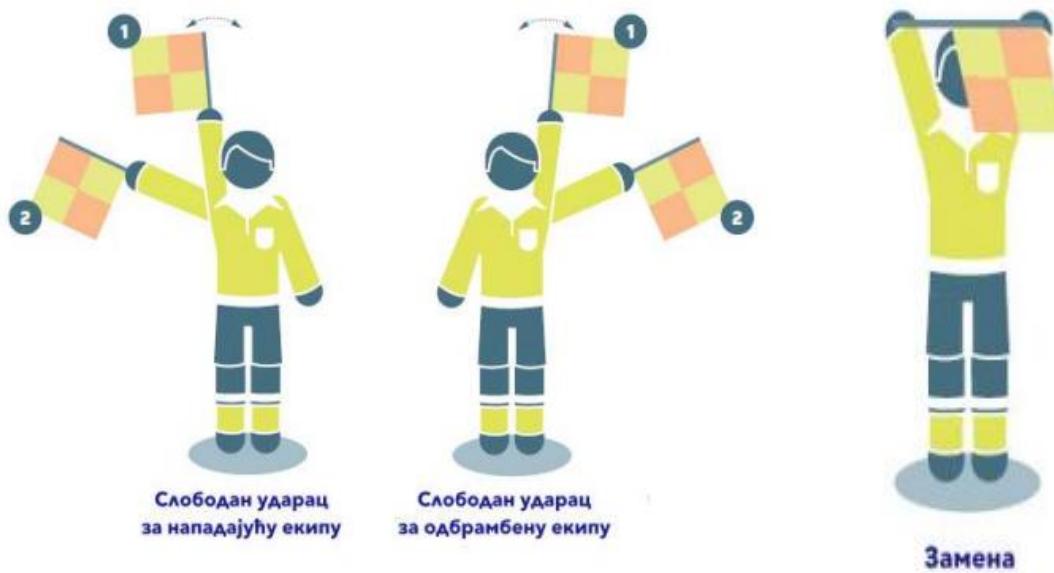
*Слика 3.1. Једноставни сигнални фудбалског судије помоћника [12]*

Сигнали који се састоје из два корака:

- Прекид – Убацивање за нападајућу страну
- Прекид – Убацивање за одбрамбену страну
- Прекид – Ударац са врата
- Прекид – Ударац из угла
- Прекид – Офсајд у близини
- Прекид – Офсајд на средини
- Прекид – удаљени офсајд
- Прекршај – Слободан ударац за нападајућу страну
- Прекршај – Слободан ударац за одбрамбену страну



Слика 3.2. Сигнализација офсајд положаја[9]



Слика 3.3. Сигнализација слободног удараца и замене[9]

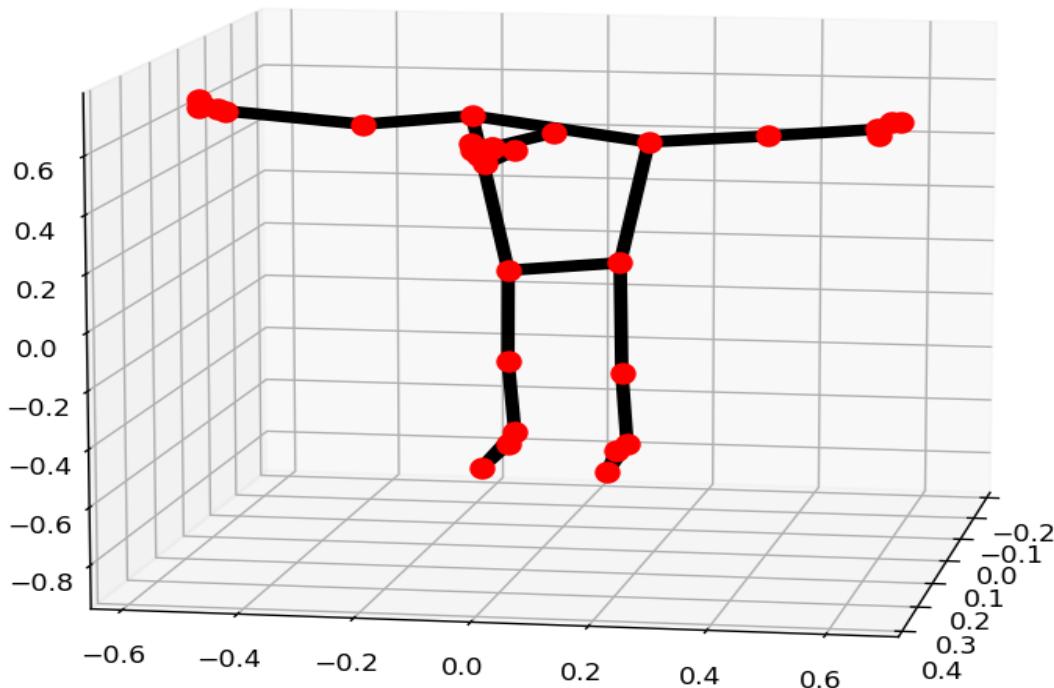
### 3.1. Идеја

Ако се пажљивије погледају сигнали фудбалског судије помоћника, могу се уочити одређени шаблони, сличности и разлике између поједињих покрета. Наравно, у закључивање треба укључити искуство, истраживање и тестирање. Заједничко свим

сигналима јесте то да судија не сме да буде у покрету, да су сви екстремитети судије помоћника исправљени, наравно, уз дозвољено одређено одступање, ноге требају да буду постављене на земљи у границама ширине кукова при свим сигналима, док се код руку ствари компликују и сви покрети имају другачије значење и захтевају другачији положај руку.

Руке у неким случајевима требају бити испред тела, у неким случајевима у равни са телом, због тога је за процену сигнализације судије помоћника искоришћена библиотека која омогућава 3D процену позе тела особе, како би се знало где се руке налазе у односу на тело. У складу са тим, обзиром да MediaPipe омогућава препознавање кључних тачака на телу, положај кључних тачака ће бити искоришћен како би се израчунато угао између одређених кључних тачака и да ли је тај положај исправан и у одређеним границама за поједине сигнале. У наставку ће сваки сигнал фудбалског судије помоћника бити објашњен.

MediaPipe омогућава презентацију положаја тела човека у 3D простору, што подразумева приказ кључних тачака на телу човека уз помоћ x, y и z координата. Како се може десити да неке тачке нису видљиве, за сваку тачку постоји и атрибут **visibility**, који даје стање о томе да ли јесте или није видљива.



*Слика 3.4. Приказ кључних тачака особе у 3D простору*

На слици 3.4. представљен је skeleton-based модел особе приказане у 3D простору који се добија коришћењем библиотеке MediaPipe када је у питању препознавање позе тела код човека (HPE). Црвеним тачкама су означене кључне тачке на телу човека (landmarks), којих има 33, прним линијама су представљене везе између зглобова на телу.

Значење координата излаза MediaPipe позе [13]:

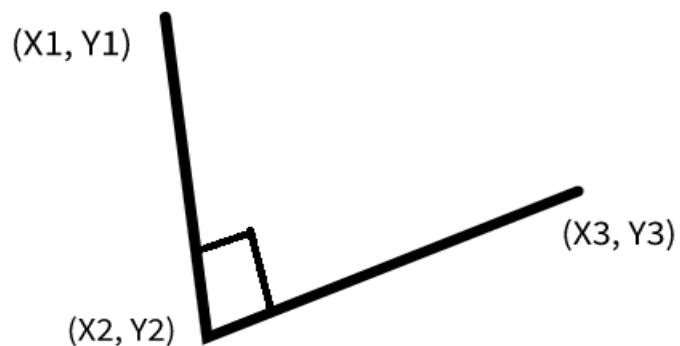
- **x** координата представља нормализовану вредност ширине фотографије у вредностима [0.0, 1.0].
- **y** координата, слично као и **x**, представља нормализовану вредност висине фотографије у границама [0.0, 1.0].
- **z** координата, представља дубину фотографије, ова координата може имати позитивне и негативне вредности, вредност ове координате је 0 на средишњој тачки између кукова, ако је вредност већа од нуле, то значи да је кључна тачка тела даља од кукова у односу на камеру, док је у супротном ближа камери.

z координата је од велике користи из простог разлога зато што се на основу ње може одредити да ли је кључна тачка ближа или даља куковима, а то је од значаја када се рачуна да ли је неки екстремитет у равни са телом тј. када је потребно израчунати угао руке судије помоћника при показивању офсајд положаја или да се провери да ли је неки екстремитет у равни са телом.

Сада постоји основа и може се размишљати о начинима како препознати да ли је поза судије помоћника исправна на основу снимка са само једне RGB камере у резолуцији погодној за MediaPipe и BlazePose алгоритам.

Како израчунати угао између произвољне 3 кључне тачке на телу човека?

Обзиром на то да се 3 тачке могу посматрати као две независне дужи које се спајају у једној тачки, тј. једна тачка ће представљати средишњу тачку која је заједничка за две дужи што је сликовито представљено на слици 3.5.:



*Слика 3.5. Поставка проблема*

Може се искористити формула за рачунање угла између два вектора:

$$\text{angle} = \text{atan2}(y_3 - y_2, x_3 - x_2) - \text{atan2}(y_1 - y_2, x_1 - x_2) [1]$$

обзиром на то да атан2 може дати вредност између  $\pi$  и  $-\pi$ , овакав израз даће вредност између  $2\pi$  и  $-2\pi$ , затим се та вредност преводи у степене, пошто ће у апликацији бити потребне само позитивне вредности, када се након примене формуле добије негативан резултат, једноставно ће се израчунати позитиван угао као:

$$\text{angle} = 360 + \text{angle}$$

Овим се добија угао између било које три кључне тачке на телу човека. Наведена формула служи за израчунавање угла у 2D равни, а обзиром да постоји и z оса, може се сматрати да се положај неког дела тела састоји из две врсте углова, један угао ће представљати угао у односу на раван коју заклапају x и y оса, док ће други угао, представљати угао који заклапају z и y оса. Први даје информацију о положају екстремитета који се налазе у равни са телом, док други даје информацију о одступању од поменуте равни.

Решен је проблем координата кључних тачака на телу, затим је решен проблем углова између кључних тачака на телу. Следећи корак јесте дефинисање правила по којима ће се знати који углови екстремитета задовољавају који покрет судије помоћника, које су граничне вредности и које вредности су прихватљиве тј. колико је дозвољено одступање, а да покрет судије буде јасан ширем и ужем аудиторијуму.

Узевши у обзир, да када би се тражио перфектан угао под којим било који екстремитет треба да се налази, се никада не би дошло до резултата, из простих разлога зато што је сама људска конституција таква и разликује се од особе до особе, затим, MediaPipe не рачуна егзактне координате већ врши процену ка координатама које ће представљати најбољи резултат, а при рачунању углова мала грешка у координатама може довести до десетак степени разлике у углу. Јасно је да ће бити потребно тестирање и усвајање одређених граница које ће бити прихватљиве и у складу са исправним сигнализирањем.

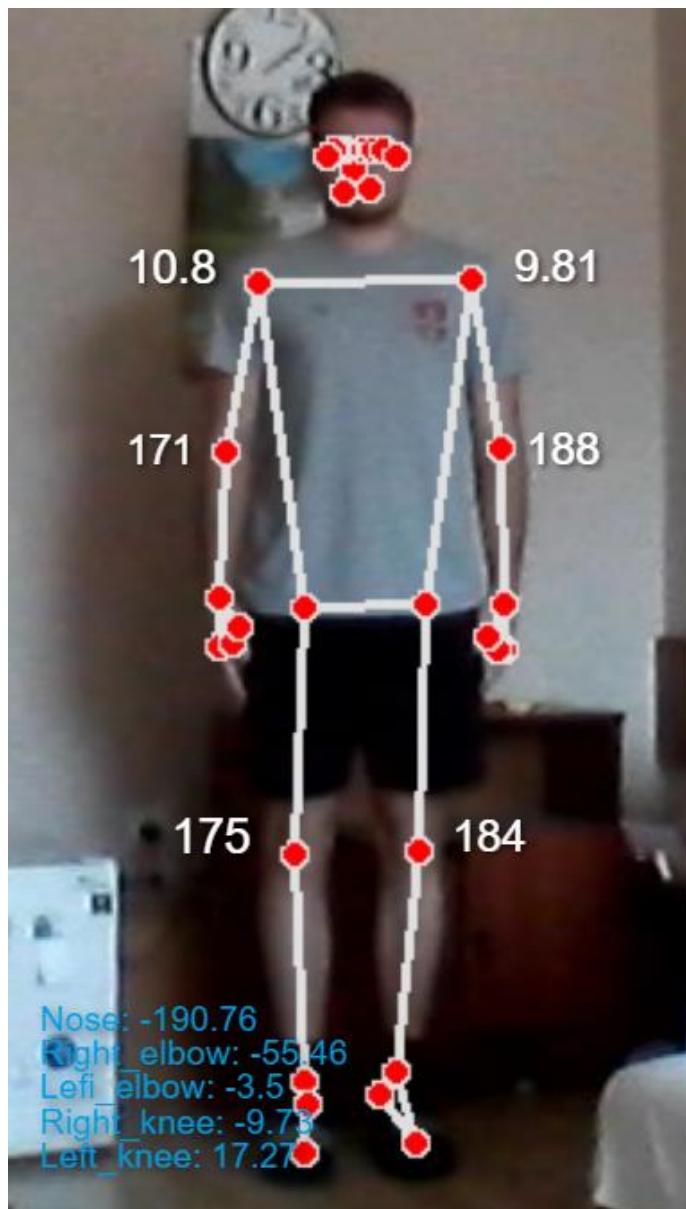
## 3.2. Класификација сигнала фудбалског судије помоћника

Као што је у тексту већ наведено, при сигнализацији неки делови тела имају једноструку улогу, док постоје делови тела који имају вишеструку улогу, у даљем тексту биће дефинисана правила по којима ће се вршити класификација сигнализације фудбалског судије помоћника.

За почетак, биће дефинисане опште ствари за све сигнале:

- Тело судије помоћника треба да мирује тј. да буде статично, осим помераја руку при показивању комбинованих покрета
- Свака од ногу треба да буде под углом приближним углу од  $180^\circ$  (за угао се узимају координате тачке на куку те ноге, затим координате тачке са колена и координате ножног зглоба, где су координате колена пресечна тачка две дужи)
- Ноге требају бити постављене у ширини кукова или мање
- Ноге требају да буду у равни са куковима тј. телом, то значи да не требају да буду испред или иза тела
- Руке требају да буду приближно под углом од  $180^\circ$  (за кључне тачке се узимају координате тачака са рамена, лакта и ручног зглоба, где средишњу тачку представљају координате лакта)
- Руке су у равни са куковима тј. телом, осим у специјалним случајевима показивања офсајда и ударца са врата (за кључне тачке се узимају кук, раме и лакат, где раме представља средишњу тачку)

Усвојено је да је рука у равни са телом, када се лакат налази не ближе камери од кључне тачке носа и не даље камери од тачке која представља супротну вредност кључне тачке носа за z-осу, исто правило важи и за ноге, кад се у обзир узму колена.



*Слика 3.6. Основне заједничке карактеристике сигнала фудбалског судије помоћника*

На слици 3.6. приказани су углови одређених делова тела и вредности z-осе кључних тачака на телу које су од значаја. Јасно је да сви углови одступају од савршеног угла од  $180^\circ$ , али сви имају приближне вредности “савршеној”.

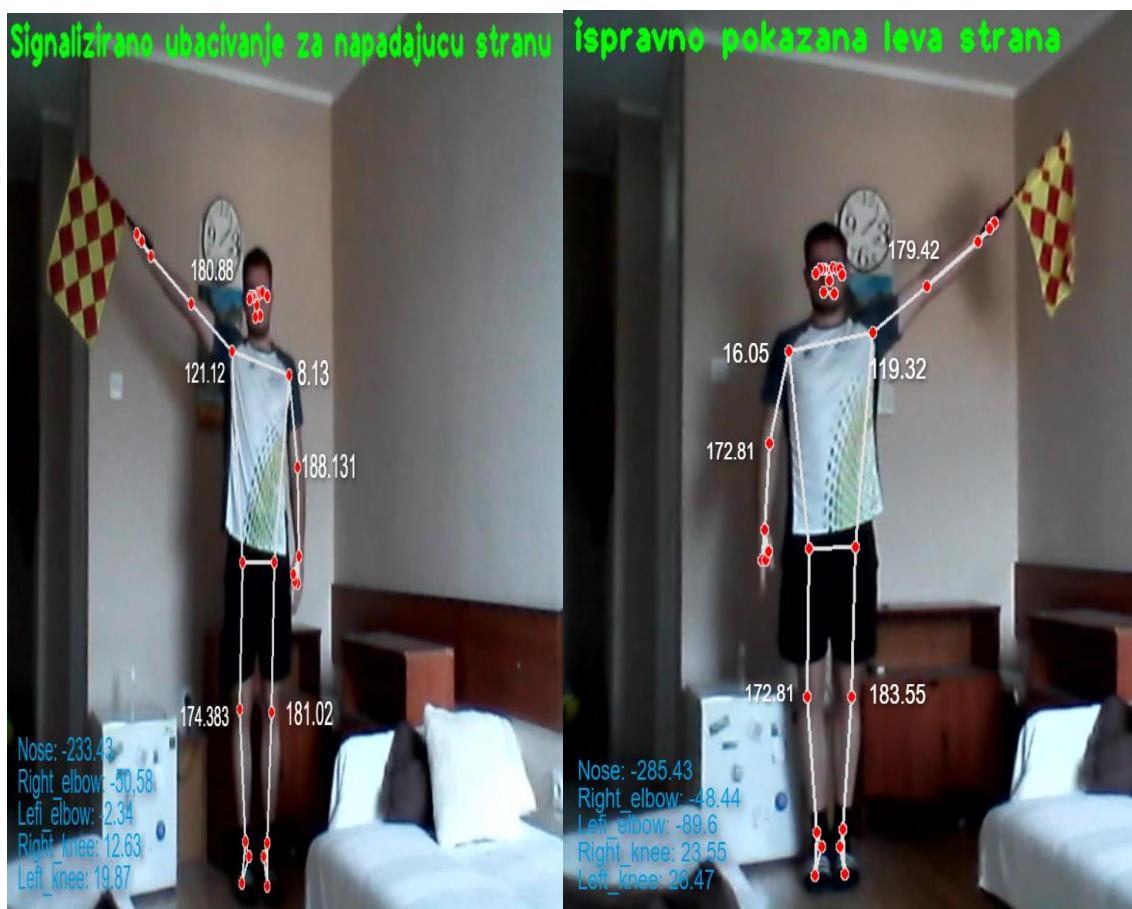
За рачунање угла код лакта, усвојене су кључне тачке са рамена, лакта и ручног зглоба, слична ситуација је и са коленом, међутим, када је реч о углу рамена, извршена је мала измена.

Наиме, када би се узеле добијене вредности за x и y координате, угао би морали да упоређујемо са другачијом вредношћу од  $180^\circ$ , зато што се кук и раме не налазе на истој позицији за x координату, па је како би се тај проблем решио примењена следећа

логика, за израчунавање угла, нама у суштини требају вредности у координате свих кључних тачака, али са х координатама можемо да манипулишемо, из тог разлога је за вредност х координате кука, при израчунавању усвојена иста вредност као и вредност х координате рамена, сада смо добили угао од нула степени и у односу на њега рачунамо покрет лакта у односу на раме.

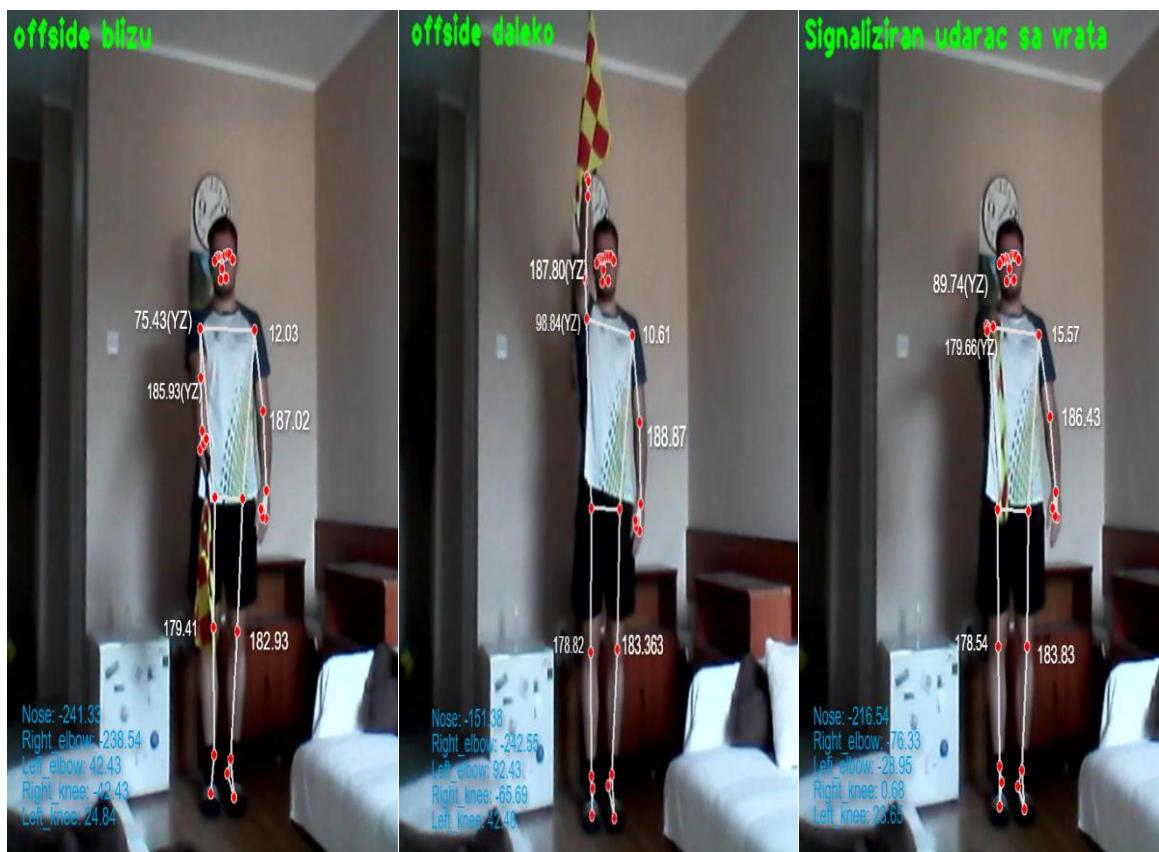
У доњем левом углу слике 3.6. Приказане су вредности “дубине” кључних тачака, тачније з координате. Из самих резултата се види да је поређење дубине кључне тачке носа и других тачака на телу, где су нама од интереса кључне тачке колена и лактова, у потпуности доволно како би се доказало да ли одређени део тела јесте или није у равни са телом, наравно уз минимална одступања која су прихватљива. Узима се у обзир и страна носа супротна у односу на z-осу, делује као да је логично да ако је испуњен услов да је део тела иза носа, онда је и са супротне стране иста ситуација, међутим не, мора се проверити да ли је супротна вредност з координате носа већа/мања од вредности кључне тачке са којом се упоређује.

Сада имамо основни положај тела судије помоћника, када се то установи и закључи који делови тела које критеријуме морају да задовољавају, може се прећи на специјализовање и класификацију значења покрета судије помоћника, тј. на препознавање разних сигнала, за почетак основних, а затим и комплексних.



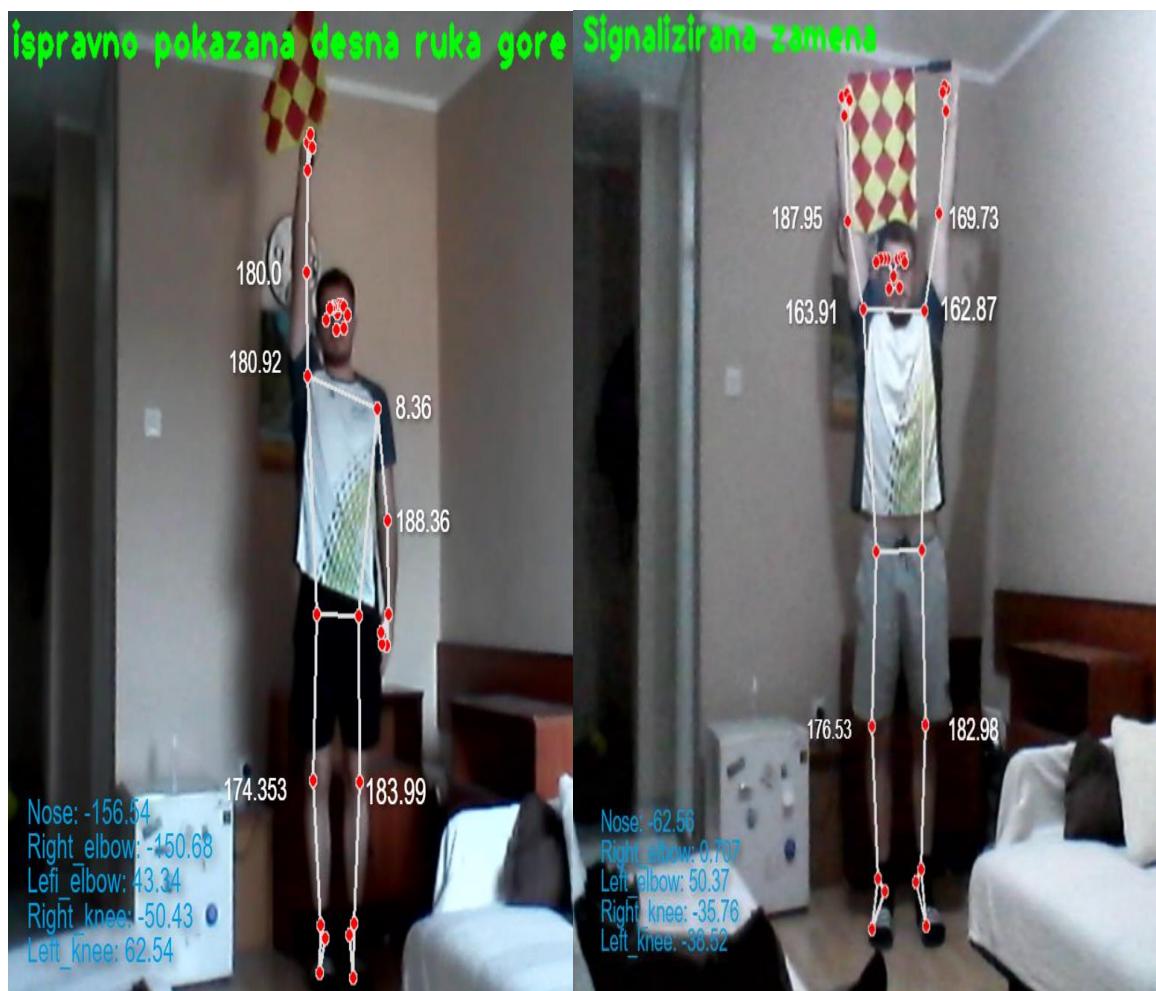
Слика 3.7. Сигнализирање стране

На слици 3.7. приказан је пример препознате позе тела за показивање којој страни ће лопта припадати. Наведена је статистика, тј. вредности углова одређених делова тела који су од интереса и “дубина” делова тела од интереса како би се приказало поређење и смисленост оваквог приступа. Углови колена су приближно  $180^\circ$ , као и лактова, тако да задовољавају услове за праве екстремитетете, затим, рамена су у положају одговарајућем за показивање траженог сигнала. Из статистике се види да су потребне кључне тачке на телу иза носа гледано у односу на z координату.



*Слика 3.8. Сигнализирање офсајда и удараца са врата*

Када је у питању сигнализирање офсајд положаја, поред основних захтева судијског сигнала, треба се испитати и угао под којим се налази рука судије помоћника, али овога пута то не зависи од x и у координата већ од z и у координата. Када је у питању офсајд на средини терена, сигнализација је иста као и код показивања удараца са вратом, само што офсајд положају предстоји сигнализација прекида, што ће бити обраћено у наставку. Важе исти услови за то да ли су делови тела од интереса испред или иза носа по z координати, међутим искључује се захтев да десни лакат мора да задовољи дати услов, јер сада он има улогу дела тела који треба да буде испред тела.



*Слика 3.9. Сигнализација прекида и замене*

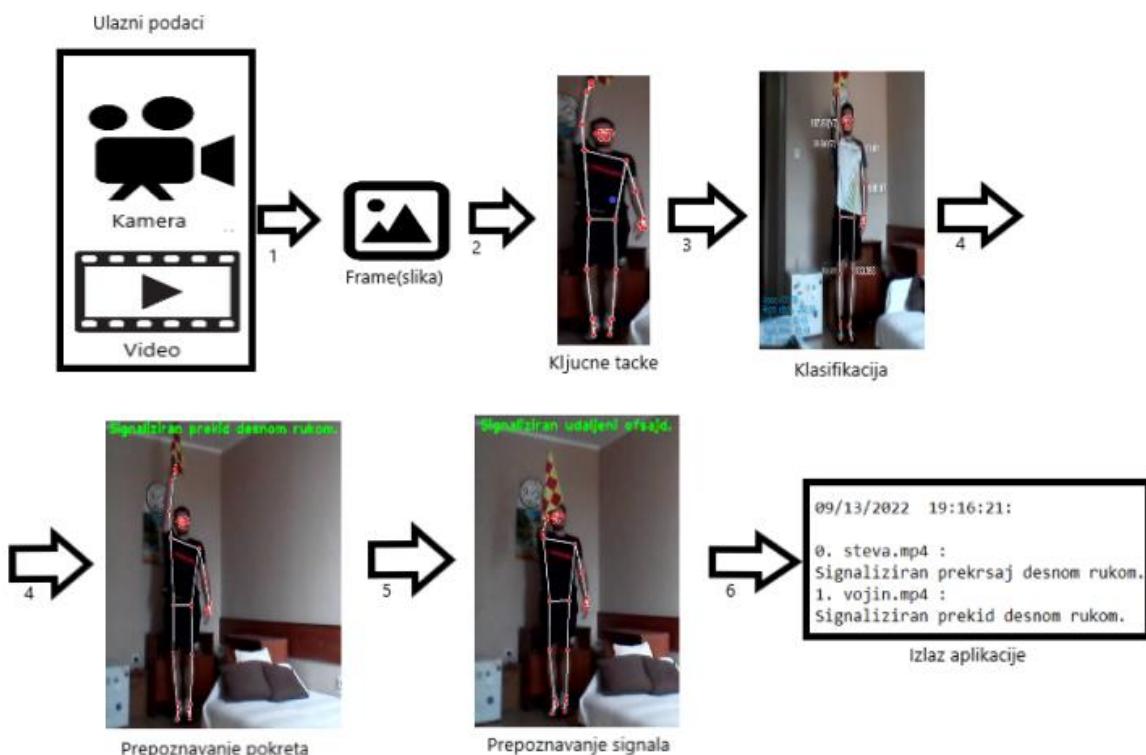
Са слике 3.9. може се видети да су прекид и замена веома слични сигнали, осим што је код једног лева рука уз тело, а код другог изнад тела као и десна. Задовољавају све основне захтеве и своје специфичне. Екстремитети требају бити у равни са телом што и јесу, па оваква показивања задовољавају услове исправних сигнала.

Конечно, све могуће опције су обрађене, следећи корак јесте комбинација ових корака, у примерима није наведена слика за сигнализирање прекршаја, због тога што је то покрет који се састоји из померања руке и његово препознавање захтева одређени алгоритам који ће бити описан у наставку. Комбинацијом наведених корака могу се описати сви сигнали фудбалског судије помоћника. Сви положаји тела који не задовољавају један од горенаведених услова и дефинисаних правила сматраће се непознатим.

## 4. Имплементација демо апликације за препознавање сигнала фудбалског судије помоћника

Након што су установљена правила по којима ће се препознавати одређени сигнали судије помоћника, могућа је имплементација демо апликације.

Апликација је креирана у Python програмском језику, основне библиотеке које су биле потребне за њену имплементацију јесу cv2 и mediapipe.



Слика 4.0. Целокупан процес рада апликације

Слика 4.0., На улазу може бити видео снимак са камере у реалном времену или већ постојећи снимак. „Хвата“ се сваки фрејм (корак 1.), са видео снимка и исти обрађује. Следећи корак у обради фотографије јесте детектовање кључних тачака на телу човека, то се ради уз помоћ функције detectPose (корак 2.), након што су познате координате кључних тачака на телу човека могу се одредити углови између кључних тачака (функција calculateAngle) од интереса и на основу тога закључити којој категорији покрета (функција poseClassification) припада тренутна поза тела (корак 3.), затим се врши препознавање покрета (корак 4.), обзиром да сигнал судије помоћника може да се састоји из више покрета, проверава се да ли тренутни покрет прати неки други покрет (корак 5.), након што је препознат сигнал судије помоћника и обрађени сви фрејмови у видео снимку, на излазу апликације се кориснику приказује резултат теста и забележени подаци смештају у фајл који води евиденцију о истим (корак 6.).

У наставку ће детаљније бити описане сви неопходни параметри и све неопходне функције које учествују у процесу препознавања сигнала фудбалског судије помоћника.

Пошто се препознаје поза на телу човека, на почетку је потребно декларисати и иницијализовати пар основних променљивих које ће бити коришћене кроз апликацију, а везане су за препознавање позе тела човека.

```
# Initializing mediapipe pose class.  
mp_pose = mp.solutions.pose  
# Initializing mediapipe drawing class, useful for annotation.  
mp_drawing = mp.solutions.drawing_utils
```

*Слика 4.1. Иницијализација mp\_pose и mp\_drawing променљивих*

**mp\_pose** - променљива која представља објекат класе **pose** из библиотеке mediapipe, ово се ради због тога што MediaPipe има огроман сет функционалности за препознавање различитих делова тела, а препознавање позе тела је само једна од функционалности, ова променљива ће бити коришћена сваки пут када се ради са позом човека и садржи све потребне атрибуте и функције за манипулацију подацима о пози човека.

**mp\_drawing** – променљива која се користи сваки пут када има потребе да се поза тела човека представи сликовито, најчешће се користи за цртање кључних тачака на телу особе, за шта је и коришћена у апликацији.

```
# Setting up the Pose function.  
pose = mp_pose.Pose(static_image_mode=False, min_detection_confidence=0.7, model_complexity=2)
```

*Слика 4.2. Подешавање Pose функције*

**Pose** библиотечка функција, иницијализује mp\_pose објекат и то је функција која дефинише начин на који ће улазна слика бити обрађена, поставка њених параметара умногоме одлучује о квалитету препознавања кључних тачака на телу и брзини обраде фотографије, наравно квалитет и брзина су у овом случају обрнуто пропорционалне величине и јасно је да се са повећањем брзине губи на квалитету.

Њени параметри:

- STATIC\_IMAGE\_MODE – дифолтно постављен на False, овај параметар функцији говори да ли се ради о видео запису или фотографији, тачније, да ли ће се поза процењивати у сваком фрејму или само када се изгуби траг кључним тачкама, False доприноси брзини обраде фотографија, док True даје прецизније координате тачака, али смањује брзину обраде узастопних фотографија. У апликацији је одабрано да буде False, из простог разлога што се обрађују видео записи.
- MODEL\_COMPLEXITY – вредности 0, 1 или 2, од најнижег нивоа комплексности обраде до највишег, респективно. Дифолтно 1, али је у апликацији одабрано да то буде 2 због што тачнијег прорачуна кључних тачака.
- SMOOTH\_LANDMARKS – Служи за смањење подрхтавања слике, дифолтна вредност True, што је и уважено.
- ENABLE\_SEGMENTATION – Ако се жели приказати и маска сегментације. False по дифолту. У апликацији није потребан приказ маске сегментације.
- SMOOTH\_SEGMENTATION – Такође служи за смањење подрхтавања маске сегментације. Дифолтно False. Овај параметар није од интереса у апликацији.
- MIN\_DETECTION\_CONFIDENCE – Минимална вредност поузданости из модела откривања особа да би се откривање сматрало успешним, узима вредности из опсега [0.0, 1.0]. Овај параметар је од великог значаја и што је већа његова вредност то је прецизније препознавање кључних тачака, овај параметар такође утиче на брзину обраде фотографија. У апликацији је тестирањем установљено да је 0.7 вредност овог параметра која даје најбоље резултате.
- MIN\_TRACKING\_CONFIDENCE – Слично као и претходни параметар, само што је у питању праћење особе. Овај параметар није од интереса у апликацији, подразумевано је постављен на 0.5.

Кључна функција за рад апликације јесте detectPose, слика 4.3., ова функција извршава детекцију позе човека са улазне слике, тј. проналази кључне тачке, процесуира их и као резултат враћа слику на којој су обележене.

```

def detectPose(image, pose):
    # Kreiranje kopije ulazne slike.
    output_image = image.copy()

    # Konvertovanje slike iz BGR u RGB format.
    imageRGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    results = pose.process(imageRGB)

    height, width, _ = image.shape
    landmarks = []

    # Ako je funkcija.
    if results.pose_landmarks:
        # Oznacavanje pronadjenih kljucnih tacaka na ulaznoj slici.
        mp_drawing.draw_landmarks(image=output_image,
                                   landmark_list=results.pose_landmarks,
                                   connections=mp_pose.POSE_CONNECTIONS)

        # Provera da li postoje kljucne tacke i ubacivanje u listu.
        for landmark in results.pose_landmarks.landmark:
            landmarks.append((int(landmark.x * width), int(landmark.y * height),
                              (landmark.z * width)))

    # Funkcija vraca obradjenu sliku i kljucne tacke.
    return output_image, landmarks

```

*Слика 4.3. Функција detectPose*

Улазна слика, по правилима OpenCV-а је у BGR формату, на почетку је потребно ту фотографију конвертовати у формат одговарајући за процесирање од стране функције process (), ова функција врши сву логику препознавања кључних тачака на телу особе и враћа резултат у виду нумеричких података о фотографији, следећи корак јесте означавање кључних тачака на слици чији су параметри фотографија на којој се цртају добијене кључне тачке, саме кључне тачке и везе између кључних тачака. Како би се даље могли обрађивати добијени подаци, они се смештају у листу landmarks, приликом смештања координате се требају превести у облик који одговара стварној размери фотографије, јер су вредности добијене функцијом process () у опсегу [0.0, 1.0]. Функција враћа излазну слику и добијене кључне тачке.

Следећа у низу функција јесте calculateAngle чији је задатак да на основу улазних података који представљају три кључне тачке од интереса, за које се треба израчунати угао који заклапају, врати угао.

```

def calculateAngle(landmark1, landmark2, landmark3):
    x1, y1, _ = landmark1
    x2, y2, _ = landmark2
    x3, y3, _ = landmark3

    # Racunanje ugla
    angle = math.degrees(math.atan2(y3 - y2, x3 - x2) - math.atan2(y1 - y2, x1 - x2))
    if angle < 0:
        angle += 360
    return angle

```

*Слика 4.4. calculateAngle*

Функција рачуна угао у 2D простору за координате x и y како би се угао могао поредити са захтевима сигнала судије помоћника. Резултат који се добија је између  $\pi$  и  $-\pi$  па се због тога негативни углови преводе у позитивне. Постоји и њена сродна функција, calculateAngleYZ чији је задатак исти, али само се уместо x координате користи z координата и ова функција служи за добијање угла за случајеве постављања делова тела испред или иза тела судије помоћника.

```

def calculatePartAnglesXY(landmarks):
    # racunanje ugla pozicije leve ruke u odnosu na: rame-lakat-zglob
    left_elbow_angle = calculateAngle(landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value], landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value],
                                       landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value])

    # racunanje ugla pozicije desne ruke u odnosu na: rame-lakat-zglob
    right_elbow_angle = calculateAngle(landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value], landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value],
                                       landmarks[mp_pose.PoseLandmark.RIGHT_WRIST.value])

    # racunanje pozicije levog ramena u odnosu na lakat-name-kuk(ugao leve ruke u odnosu na telo)
    left_shoulder_angle = calculateAngle(landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value],
                                          landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value],
                                          # problem je sto kuk i rame nisu u liniji, a potreban mi je taj ugao da bude 0 kako bih racunao prav
                                          (landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value][0],
                                           landmarks[mp_pose.PoseLandmark.LEFT_HIP.value][1],
                                           landmarks[mp_pose.PoseLandmark.LEFT_HIP.value][2]))

    # racunanje pozicije desnog ramena u odnosu na lakat-name-kuk(ugao desne ruke u odnosu na telo)
    right_shoulder_angle = calculateAngle((landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value][0],
                                            landmarks[mp_pose.PoseLandmark.RIGHT_HIP.value][1],
                                            landmarks[mp_pose.PoseLandmark.RIGHT_HIP.value][2]),
                                           landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value],
                                           landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value])

    # racunanje pozicije levog kolena u odnosu na: kuk-koleno-zglob
    left_knee_angle = calculateAngle(landmarks[mp_pose.PoseLandmark.LEFT_HIP.value],
                                      landmarks[mp_pose.PoseLandmark.LEFT_KNEE.value],
                                      landmarks[mp_pose.PoseLandmark.LEFT_ANKLE.value])

    # racunanje pozicije desnog kolena u odnosu na: kuk-koleno-zglob
    right_knee_angle = calculateAngle(landmarks[mp_pose.PoseLandmark.RIGHT_HIP.value],
                                      landmarks[mp_pose.PoseLandmark.RIGHT_KNEE.value],
                                      landmarks[mp_pose.PoseLandmark.RIGHT_ANKLE.value])

    return left_elbow_angle, right_elbow_angle, left_shoulder_angle, right_shoulder_angle, left_knee_angle, right_knee_angle

```

*Слика 4.5. Функција calculatePartAnglesXY*

На слици 4.5. може се видети функција која се користи како би се израчунали углови делова тела од интереса, наиме, овде се врши позивање функције са слике 4.4. и постављају потребни параметри, у већини случаја се прослеђују координате кључних тачака добијене из функције detectPose, међутим у неким случајевима је потребна

минимална модификација, наиме, када је потребно рачунати угао између лакта, рамена и кука, како би се добио угао који је између  $0^\circ$  и  $180^\circ$ , пошто раме и кук нису у истој равни гледано у односу на x-осу, а у апликацији је потребна информација о томе под којим углом се рука налази у односу на тело, за вредност x координате кука ће се узети вредност x координате рамена. На тај начин је добијена права представа и тачна вредност угла који је потребан за испитивање. Повратна вредност јесу углови свих екстремитета и делова од интереса гледано у односу на раван xy.

Ова функција има себи сродну функцију calculatePartAnglesYZ, чији је задатак исти, само што се у овом случају позива функција calculateAngleYZ и враћа вредности углова делова тела гледано у односу на уз раван, то су случајеви када се испитује угао офсајд положаја или случајеви када се испитује да ли је рука или нога уз тело, тј. да ли задовољава основне услове сигнала судије помоћника.

За препознавање позе судије помоћника је потребно испитати доста параметара, функција у којој се то испитује јесте classifyPose, њен задатак је да на основу улазних параметара, а то су координате кључних тачака и улазна слика, одреди у ком се положају налази тело судије помоћника, обзиром да постоји доста валидних положаја судије помоћника, ова функција је веома комплексна и много услова треба бити истовремено испуњено, док је с друге стране доста сличних правила са једном променом које одлучују о томе у ком је положају судија помоћник. Наравно, сваки пут када поза не задовољава потребне услове за било који сигнал, повратна вредност ће бити “Nepoznata pozna”.

```
#provera da li su obe noge ispravljene
if ((left_knee_angle > 165 and left_knee_angle < 195 and right_knee_angle > 165 and right_knee_angle < 195) and
    (landmarks[mp_pose.PoseLandmark.RIGHT_KNEE.value][2]>landmarks[mp_pose.PoseLandmark.NOSE.value][2] and
     landmarks[mp_pose.PoseLandmark.RIGHT_KNEE.value][2]<-landmarks[mp_pose.PoseLandmark.NOSE.value][2]) and
    (landmarks[mp_pose.PoseLandmark.LEFT_KNEE.value][2]>landmarks[mp_pose.PoseLandmark.NOSE.value][2] and
     landmarks[mp_pose.PoseLandmark.LEFT_KNEE.value][2]<-landmarks[mp_pose.PoseLandmark.NOSE.value][2])):
    # provera da li su leva i desna ruka ispravljene/razlika kada je ruka napred i u stranu je xy i yz
    if left_elbow_angle > 160 and left_elbow_angle < 210 and \
        ((right_elbow_angle > 160 and right_elbow_angle < 210) or (right_elbow_angleYZ > 160 and right_elbow_angleYZ < 190)):
```

*Слика 4.6. Основна класификација позе*

Основни услов јесте да су обе ноге исправљене и да су обе руке исправљене, разлика је једино у томе што лева рука никада неће имати потребе да буде исправљена у односу на z осу, тако да се за њу дозвољава даља обрада само ако је у равни са x осом.

Након установљавања главних услова, приступа се провери услова који требају бити задовољени за тачност сигнализирања судије помоћника (Слика 4.7.), даља разлика јесте у сигналима који се показују са рукама у равни са x-осом и сигналима који се показују у равни са y-осом.

```

#provera da li su ruke u ravni sa bocnim stranama tela
if (((landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value][2]>landmarks[mp_pose.PoseLandmark.NOSE.value][2] and
      landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value][2]<-landmarks[mp_pose.PoseLandmark.NOSE.value][2]) or right_shoulder_angleYZ > 110) and
      (landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value][2]>landmarks[mp_pose.PoseLandmark.NOSE.value][2] and
       landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value][2]<-landmarks[mp_pose.PoseLandmark.NOSE.value][2])):
    #provera da li je leva ruka uz telo
    if left_shoulder_angle > 0 and left_shoulder_angle < 30:
        #provera da li je desna ruka odrucena ispravno za pokazivanje auta i preksaja
        if right_shoulder_angle > 100 and right_shoulder_angle < 145:
            label="Signalizirana desna strana."
        # provera da li je desna ruka odrucena ispravno za pokazivanje udarca iz ugla
        if right_shoulder_angle > 30 and right_shoulder_angle < 75:
            label = "Signaliziran udarac iz ugla."
        # provera da li je desna ruka odrucena ispravno za pokazivanje ofsaida
        if right_shoulder_angle > 170 and right_shoulder_angle < 190:
            label = "Signaliziran prekid desnom rukom."
    # provera da li je desna ruka uz telo
    if right_shoulder_angle > 0 and right_shoulder_angle < 30:
        # provera da li je leva ruka odrucena ispravno za pokazivanje auta i preksaja
        if left_shoulder_angle > 100 and left_shoulder_angle < 145:
            label = "Signalizirana leva strana."
        if left_shoulder_angle > 170 and left_shoulder_angle < 185:
            label = "Signaliziran prekid levom rukom."
    if right_shoulder_angle > 150 and right_shoulder_angle < 185 and left_shoulder_angle > 150 and left_shoulder_angle < 185:
        label = "Signalizirana zamena."

```

*Слика 4.7. Класификација сигнала када су руке у равни са x-осом*

Прва ставка је провера да ли су руке у равни са телом, ако је то испуњено врши се провера углова делова тела од интереса, у овом случају, су то руке, најдубљи ниво провере сигнала, постоје сигнали десном, левом и обема рукама, наравно када се показује сигнал десном руком, лева рука мора да мирује и обрнуто, покрети се на крају разврставају по угловима у којима се налази рука.

Ако услов за руке у равни са телом није испуњен, проверава се да ли судија помоћник можда сигнализира офсајд положај (Слика 4.8.).

```

#offside
# provera da li je leva ruka uz telo za offside
if (left_shoulder_angle > 0 and left_shoulder_angle < 30 and right_shoulder_angleYZ < 120
    and landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value][2]<landmarks[mp_pose.PoseLandmark.NOSE.value][2]
    and (Landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value][2]>landmarks[mp_pose.PoseLandmark.NOSE.value][2] and
    landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value][2]<-landmarks[mp_pose.PoseLandmark.NOSE.value][2])):
# provera da li je desna ruka odrucena ispravno za pokazivanje ofsalda u daljini
if right_shoulder_angleYZ > 92 and right_shoulder_angleYZ < 110:
    label = "Signaliziran daleki ofsajd."
# provera da li je desna ruka odrucena ispravno za pokazivanje ofsalda u daljini
if right_shoulder_angleYZ > 30 and right_shoulder_angleYZ < 85:
    label = "Signaliziran ofsajd u blizini."
# provera da li je desna ruka odrucena ispravno za pokazivanje ofsalda u daljini
if right_shoulder_angleYZ > 85 and right_shoulder_angleYZ < 92:
    label = "Signaliziran ofsajd na sredini."

```

*Слика 4.8. Класификација сигнала када су руке у равни са z-осом*

До сада је описано одређивање статичних покрета судије помоћника, који не зависе од тога да ли је у питању видео или слика и одређују статичне покрете. Потребно је одредити и једну врсту динамичког покрета, а то је сигнализирање прекршаја, који је комбиновани покрет.

```

def isHandStill(list, flag_in_hand):

```

*Слика 4.9. Параметри функције isHandStill*

Постоји још једна врста провере сигнала, која није у classifyPose функцији, већ у главној функцији, основни разлог је то што захтева додатну помоћну функцију како би се одредило да ли судија “маше” руком, тачније да ли сигнализира прекршај. Функција која проверава да ли је рука у мировању или “маше” јесте isHandStill, њени улазни параметри јесу низ који се састоји од кључних тачака двадесет забележених фрејмова видео записа и параметра који говори која рука је подигнута. Пошто се ради о видео запису, прво проверава која је рука у питању, затим проверава последњи угао последњег фрејма који се односи на угао лакат-раме-кук, како би проверила да ли је можда покушан неки други покрет у међувремену, из разлога што за одређени покрет рука мора да буде у истом положају минимум двадесет фрејмова, на пример, рука је подигнута горе десет фрејмова, али је то било само привремено и судија спушта руку, то се класификује као невалидан потез. Ако је рука померена и покушан други сигнал, аутоматски се одбацује опција да рука није мирна и враћа вредност Фалс, која главној функцији говори да је рука у мировању и да настави даље са препознавањем покрета. Након основне провере иде израчунавање угла између лакта, ручног зглоба и малог прста. Овај угао нам може помоћи у откривању тога да ли се рука помера или је мирна, тако што се за све фрејмове израчуна угао, а затим фрејм са највећим и фрејм са најмањим углом упореде. Ако је разлика већа од  $15^\circ$ , што није случајна вредност, већ вредност која је установљена тестирањем и показало се да ако постоји разлика већа од  $15^\circ$ , у највећем броју случајева се ради о томе да рука није у стању мировања. Позив ове функције се врши на сваких двадесет фрејмова у случајевима када је рука подигнута усправно горе, овај поступак

позивања ће бити описан у главној функцији програма која користи резултате које враћа функција isHandStill.

До сада су описане све помоћне функције апликације. Сва логика саме апликације је урађена у главној функцији start (files) (Слика 4.10.), ова функција прима улазне видео записи, један или више и обрађује их, а на свом излазу поставља податке у одређену форму за приказ на страни корисника и уписује их у фајл који служи за вођење евиденције о позивима функције и њеним резултатима.

```
for index, i in enumerate(files):
    camera_video = cv2.VideoCapture(i)
    cv2.namedWindow('Procenjivanje poze', cv2.WINDOW_NORMAL)
    first_move = False
    first_move_label = ''
    second_move = False
    second_move_label = ''
    move_meaning = ''
    frame_counter = 0
    frame_counter_label = ''
    last_frame_counter = 0
    frame_check_count = 0
    frame_list = []
    two_step_move = False
    two_step_move_attempt = False
    flag_in_hand = "right"
    handNotStill = False
    result+= "\n"+ str(index) + ". "+ i.rsplit('/', 1)[-1]+": "
    while camera_video.isOpened():
        ok, frame = camera_video.read()
        if not ok:
            continue
        frame_height, frame_width, _ = frame.shape
        frame = cv2.resize(frame, (int(frame_width * (640 / frame_height)), 640))
        frame, landmarks = detectPose(frame, pose)
        label=''
        if landmarks:
            frame, label = classifyPose(landmarks, frame)
```

Слика 4.10. Декларација помоћних променљивих и позивање основних функција

Судијски сигнали могу бити разноврсни, али исто тако, на улазу се могу добити различити видео снимци, на некима судија сигнализира једноставан прекршај, на некима комбиновани, с друге стране, могући су видео снимци на којима судија показује више узастопних сигнала, због тога је потребно доста помоћних променљивих како би се тачно могли одредити покрети судије помоћнику када је у питању један или више и када се они смењују на истом видео снимку.

На почетку функције се налази for петља која служи за случај обраде више видео снимака, отвара се прозор за процењивање позе.

Значења променљивих:

- **first\_move** и **first\_move\_label** – обзиром на то да се сигнал може састојати из комбинације покрета, идеја је да се памти први и евентуално други покрет ако постоји. first\_move означава да се десио први покрет, а његова лабела памти који се сигнал дододио.
- **second\_move** и **second\_move\_label** – за разлику од first\_move\_label, вредност second\_move\_label може бити само вредност оних покрета који задовољавају услов да буду други покрет у комбинацији покрета судије помоћника, second\_move не може бити True ако first\_move није True.
- **move\_meaning** – када имамо комбинацију покрета, на основу првог и другог покрета се установљује који је сигнал судија показао, вредност иде у move\_meaning.
- **frame\_counter** и **frame\_counter\_label** – променљиве које служе за бројање понављања одређеног сигнала, наиме, пошто се сматра да је сигнал показан ако се установи да се поновио у двадесет узастопних фрејмова, ова променљива нам омогућава да знамо број понављања. frame\_counter\_label служи за упоређивање старог и новог сигнала, ако је сигнал исти ова променљива се повећава.
- **last\_frame\_counter** – променљива која броји сваки фрејм и служи за излазак из while петље, тј. прекидање обраде видео записа.
- **frame\_check\_count** и **frame\_list** – променљиве које иду у пакету и служе за припрему фрејмова за проверу да ли се рука помера или је стабилна.
- **two\_step\_move** и **two\_step\_move\_attempt** – променљиве које означавају да ли је покушан сигнал који се састоји из комбинације сигнала, two\_step\_move\_attempt служи да назначи да је покушана сигнализација прекида или прекршаја
- **flag\_in\_hand** и **handNotStill** – означавају којом руком је покушан прекид и да ли је рука мирна или не.
- **result** – променљива у којој се памте резултати препознавања позе тела, памти само препознате сигнале фудбалског судије помоћника и служи за процесирање резултата.

Затим се учитава видео снимак и чита фрејм по фрејм, величина фрејма се прилагођава величини прозора и позивају основне функције за детекцију и класификацију позе.

Функција за детекцију позе као параметре узима тренутни фрејм и већ дефинисан начин обраде позе, њен резултат се смешта у променљиве и спреман је за даљу обраду, затим се добијена фотографија и кључне тачке за одређени фрејм прослеђују функцији за класификовање сигнала, од резултата ове функције зависи како ће сигнал судије помоћника бити препознат, сада имамо кључне тачке и следећи корак јесте препознавање покрета, дефинисање да ли је комбинација покрета у питању или пак једноставан сигнал. Узето је у обзир да видео запис може садржати више узастопних сигнала судије помоћника и у складу са тим формирана логика која препознаје како једноструке, тако и

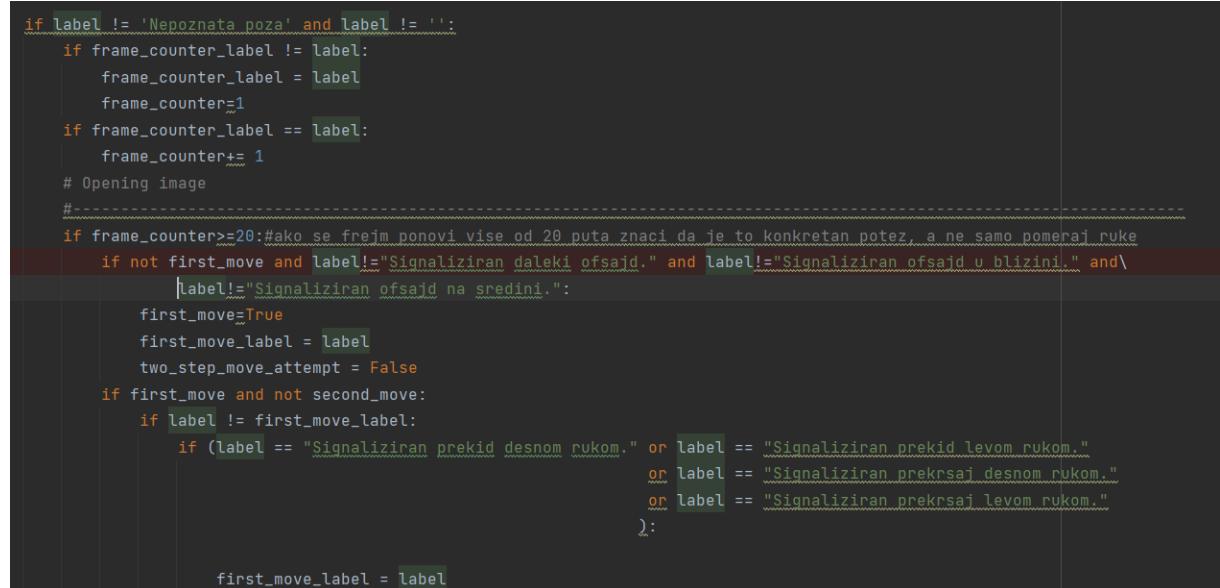
комбинације покрета, али води рачуна и о томе да ли се неки покрети понављају и не дозвољава покретима који могу да буду само први покрет да направе пометњу при препознавању сигнала.

```
if label == "Signaliziran prekid desnom rukom." or label == "Signaliziran prekid levom rukom.":  
    two_step_move = True  
    two_step_move_attempt=True  
    if label == "Signaliziran prekid levom rukom.":  
        flag_in_hand = "left"  
    if label == "Signaliziran prekid desnom rukom.":  
        flag_in_hand = "right"  
  
if two_step_move:  
    frame_check_count += 1  
    frame_list.append(landmarks)  
  
if frame_check_count == 20:  
    handNotStill = isHandStill(frame_list, flag_in_hand)  
    two_step_move = False  
    frame_check_count = 0  
    frame_list.clear()  
  
if handNotStill:  
    if flag_in_hand == "right":  
        label = "Signaliziran prekršaj desnom rukom."  
    if flag_in_hand == "left":  
        label = "Signaliziran prekršaj levom rukom."
```

Слика 4.11. Препознавање сигнализације прекријаја

Као што је раније наведено, сигнализација прекршаја има специфичан начин препознавања. Наиме, да би се знало да ли је прекршај потребно је да се деси одређени след догађаја. Прво, сваки пут када рука буде подигнута право горе, забележава се тај фрејм и рука која је подигнута, од тог тренутка креће бројање наредних 20 фрејмова и забележавање сваког од њих, то се ради због тога што мора бити позвана функција isHandStill (frame\_list, flag\_in\_hand), када се постигне тражени број фрејмова, испитује се да ли се рука особе помера или не, ако се рука помера, а притом није промењен положај руке, тј. и даље је подигнута право горе, закључује се да је сигнализиран прекршај, овај сигнал се сматра као први покрет у комбинованом сигналу и само тако може да буде обрађиван, ако се сигнал још једном понови, сматра се да је то исти сигнал, све док се не појави други сигнал који припада групи сигнала који су пратећи за прекршај. У супротном, ако се установи, да након двадесет фрејмова, рука мирује, то значи да је ипак сигнализиран прекид игре, он се обрађује на други начин. Такође, све

док се не установи да ли је прекид или прекршај, рачуна се да се десио покушај прекида, што није непозната поза, али није од значаја, јер то не мора нужно да буде сигнал.



```
if label != 'Nepoznata pozna' and label != '':
    if frame_counter_label != label:
        frame_counter_label = label
        frame_counter=1
    if frame_counter_label == label:
        frame_counter+= 1
    # Opening image
    #
    if frame_counter>=20:#ako se frejm ponovi vise od 20 puta znaci da je to konkretni potez, a ne samo pomeraj ruke
        if not first_move and label!="Signaliziran daleki ofsajd." and label!="Signaliziran ofsajd u blizini." and \
            label!="Signaliziran ofsajd na sredini.":
            first_move=True
            first_move_label = label
            two_step_move_attempt = False
        if first_move and not second_move:
            if label != first_move_label:
                if (label == "Signaliziran prekid desnom rukom." or label == "Signaliziran prekid levom rukom."
                    or label == "Signaliziran prekršaj desnom rukom."
                    or label == "Signaliziran prekršaj levom rukom."):
                    :
first_move_label = label
```

Слика 4.12. Логика првог сигнала

Наиме, за разлику од бројача који броји све фрејмове, frame\_counter броји само оне фрејмове који су узастопни и који чине одређени препознати покрет, као што је већ наведено, да би били сигурни да је покрет сигнал, а не само померај руке, потребно је да се понови на одређеном броју узастопних фрејмова, у случају ове апликације установљено је да је најоптималнији број двадесет. Пошто између фрејмова може доћи до промене покрета, креира се лабела која ће чувати покрет који се броји, ако се тај покрет промени мења се и лабела и рестартије бројач, али ако бројач стигне до траженог броја, тада се сматра да је тај покрет сигнал и он улази у даље процесирање, ако је вредност first\_move постављена на False, то значи да је то први сигнал, први сигнал може бити коначни и може бити део комплекснијег сигнала, постоји група сигнала која се сматра првим покретом, а то су прекиди, прекршаји и замене, сви остали сигнали могу постати други покрет у сигналу, али ови покрети ће се смењивати ако се мењају или остајати исти ако се не мењају, али ће увек бити први покрет. (Слика 4.12.)

```

if first_move and not second_move:
    if label != first_move_label:
        if (label == "Signaliziran prekid desnom rukom." or label == "Signaliziran prekid levom rukom."
            or label == "Signaliziran prekršaj desnom rukom."
            or label == "Signaliziran prekršaj levom rukom."):
            ):

            first_move_label = label
        elif (label=="Signaliziran daleki ofsađ." or label=="Signaliziran ofsađ u blizini." or label=="Signaliziran ofsađ
            if label=="Signaliziran ofsađ na sredini.":
                first_move_label="Signaliziran udarac sa vrata."
            else:
                result += "\n" + first_move_label
                first_move=False
                second_move=False
            elif label == "Signalizirana zamena.":
                result += "\n" + first_move_label
                first_move = False
                second_move = False
            else:
                second_move = True
                second_move_label = label

```

Слика 4.13. Логика другог сигнала

```

if second_move:
    label_holder = label
    label = move_meaning
    if len(result) < len(move_meaning) or result[(-len(move_meaning)):] != move_meaning:
        result += "\n" + move_meaning
    if second_move_label != label_holder:
        first_move = False
        second_move = False
color = (0, 255, 0)

```

Слика 4.14. Логика преласка на следећи сигнал

На сликама 4.13. и 4.14. приказана је логика другог сигнала, као што је већ наведено, други сигнал могу бити само одређени покрети. Зна се да је настао други сигнал када након првог сигнала који је један од сигнала који претходе другом активан и када се променио сигнал на неки од сигнала од интереса за други покрет. Ако сигнал који се појавио није од интереса, аутоматски се забележава да акција није имала смисла и у резултату записује да је серија претходних сигналова бесмислена, тј. да је показан само део сигнала.

Када се деси други сигнал, забележава се који је то сигнал и практично чека да се промени како би се кренуло у препознавање нових сигналова, наравно, забележен је успешан комбиновани сигнал пре наставка препознавања. Ресетују се параметри који показују присуство сигналова.

Ако је неки сигнал валидан, мења се боја лабеле за приказ сигнала на екрану, ово је од користи када се уживо прати рад програма, јер се јасно види ток препознатих

сигнала и покрета. Црвеном бојом се означава да сигнал није препознат, плавом да је покушан неки прекид или прекршај, а зеленом да је исправно показан неки сигнал. Лабела се наравно штампа на екрану и евидентно је у сваком тренутку у ком је статусу препознавање покрета судије помоћнику на датом видео снимку који је тренутно у оптицају.

```

if label == "Nepoznata poza" or label=='':
    if handNotStill:
        color = (0, 255, 0)
    else:
        label="Nepoznata poza"
        color=(0, 0, 255)
# if first_move and not second_move and label=="Signaliziran ofsajd na sredini.":

#label="Signaliziran udarac sa vrata"
if first_mo PEP 8: E265 block comment should start with '#'
    label="Signaliziran udarac sa vrata"
    color = (255, 0, 0)
    : in daleki ofsajd."
    #label="Signaliziran ofsajd u blizini" or label=="Signaliziran ofsajd na sredini.":

    label="Reformat the file Alt+Shift+Enter More actions... Alt+Enter"
    color = (255, 0, 0)
if two_step_move_attempt and not first_move:
    label="Neuspesan pokusaj prekida"
    color = (255, 0, 0)
if (label == "Nepoznata poza" or label == '') and first_move:
    if second_move:
        label = move_meaning
        color = (0, 255, 0)
    else:
        label=first_move_label
        color=(0,255,0)
if not first_move and not second_move:
    label = "Nepoznata poza"
    color = (0, 0, 255)

```

*Слика 4.15. Иницијализација приказа и приказ*

Јасно је да је приказ веома битан фактор како би се могао пратити процес препознавања сигнала судије помоћнику на самом видео запису. Због тога се обрађују и случајеви који се могу десити, а да поремете разумевање онога који гледа сам процес препознавања. Пита се да ли је рука мирна, из разлога што се може десити да је сигнализиран прекршај, а да је рука излазила из оквира право подигнуте руке на горе, па тај услов боји лабелу која носи стринг који каже да је дошло до прекршаја, у супротном је поза непозната.

Ако тренутни фрејм није препознат, а већ постоји први потез, приказује се постојећи потез, слично и за други потез. Такође остали сигнални офсајда који нису други покрет значиће да је сигнал невалидан, али да је покушана сигнализација офсајда, па је због тога лабела плаве боје.

На крају се врши приказ сваког фрејма и боји текст у одговарајућу боју.

```

k = cv2.waitKey(1) & 0xFF

last_frame_counter += 1
# Check if 'ESC' is pressed.
if (k == 27 or last_frame_counter == camera_video.get(cv2.CAP_PROP_FRAME_COUNT)):
    # Break the loop.
    break

# Release the VideoCapture object and close the windows.
camera_video.release()
cv2.destroyAllWindows()

```

*Слика 4.16. Логика гашења камере*

Када је последњи фрејм камере излази се из петље и прелази на следећи видео запис.

```

f = open("results.txt", "a")
now = datetime.now()
current_time = now.strftime("%m/%d/%Y")
current_time += now.strftime(" %H:%M:%S")
f.write("-----I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-I-----"
       "\n\n"+current_time+":\n"+result)
f.close()
label_result.configure(text=result)

```

*Слика 4.17. Упис резултата у фајл и приказ резултата кориснику*

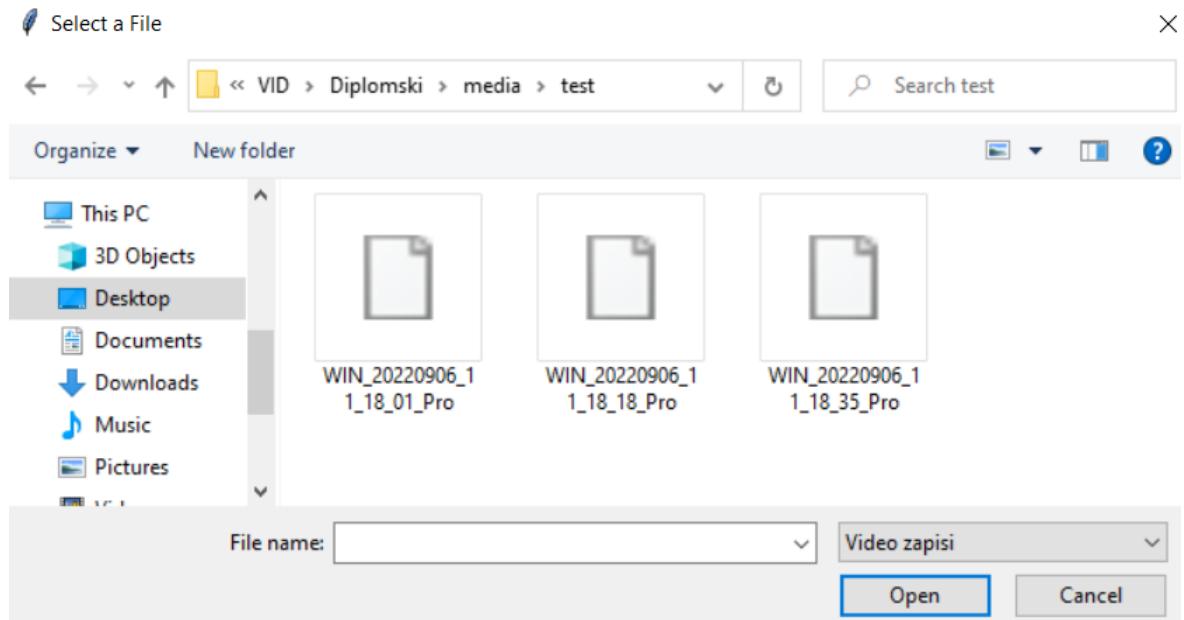
У фајлу “results.txt” се води евиденција о извршеним тестирањима и њиховим резултатима.



*Слика 4.18. Кориснички интерфејс*

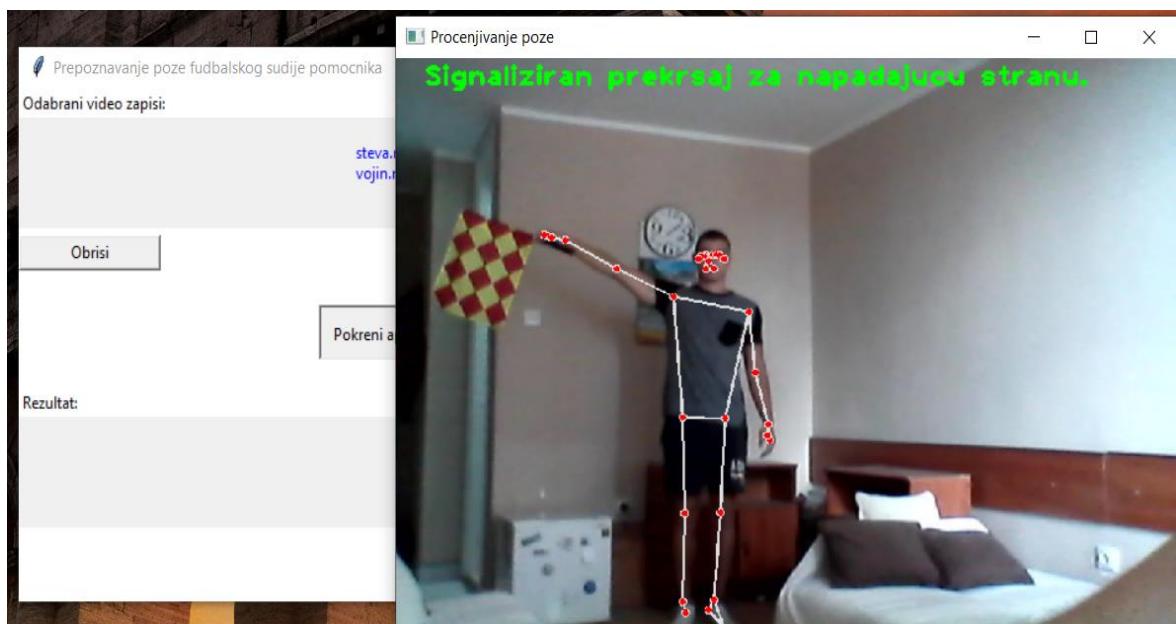
На слици 4.18. приказан је кориснички интерфејс, бројевима од 1 до 6 су означене компоненте форме:

1. Дугме намењено за брисање одабраних видео записа који су унети у лабели која је означена бројем 5 на слици.
2. Дугме које води у претрагу видео записа на рачунару, може се више пута бирати видео или се одједном бирати више видео записа.

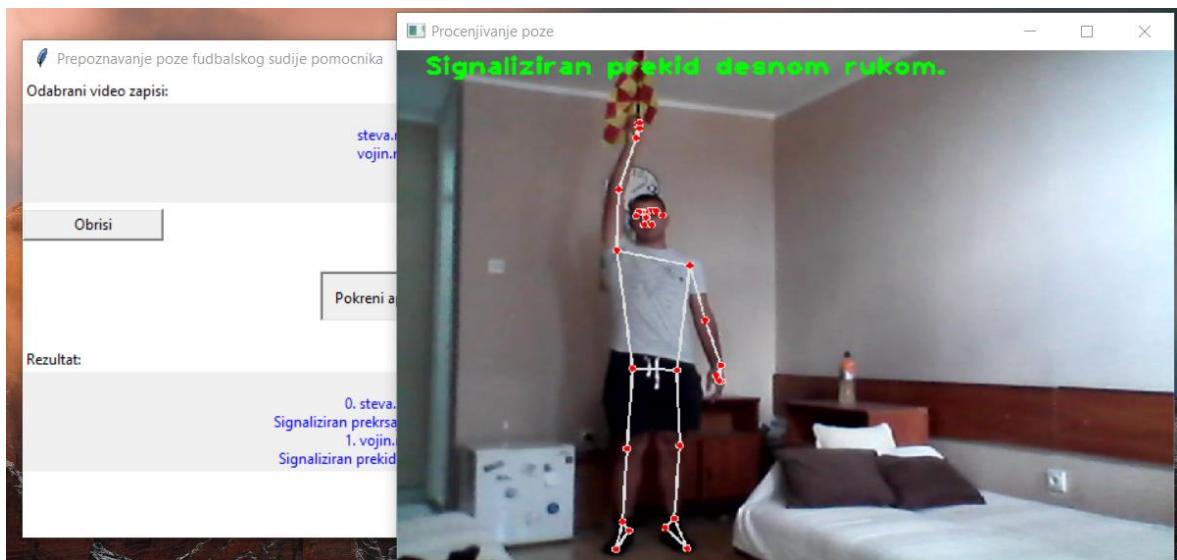


Слика 4.19. Одабир видео записа

3. Дугме за покретање апликације, кликом на ово дугме позива се функција start (files) и покреће се обрада одабраних видео записа.

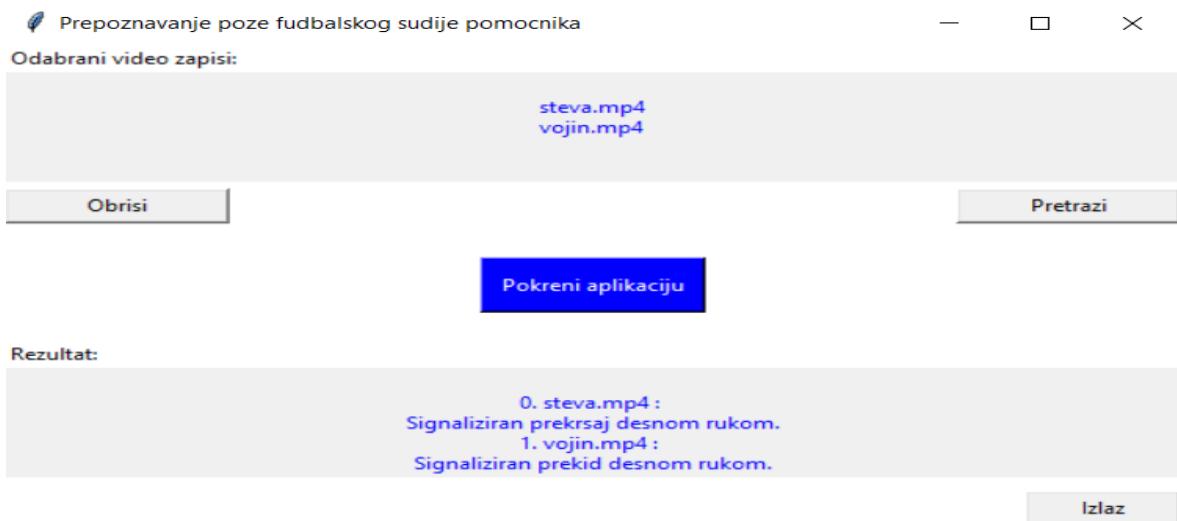


Слика 4.20. Покретање и рад апликације



Слика 4.21. Покретање и рад апликације

4. Дугме за излаз из програма.
5. Лабела за приказ одабраних видео записа за обраду
6. Лабела за приказ резултата



Слика 4.22. Изглед корисничког интерфејса након обраде фотографија.

09/13/2022 19:16:21:

0. steva.mp4 :  
Signaliziran prekrasaj desnom rukom.  
1. vojin.mp4 :  
Signaliziran prekid desnom rukom.

Слика 4.23. Подаци у фајлу за чување података о раду програма

## 5. Закључак

Детекција људске позе је проблем којим се рачунарски вид бави већ тридесетак година, из године у годину представљана су нова решења и надоградње постојећих како би се дошло до перфектних резултата, савршенство ни дан данас није постигнуто, али су данашње технологије на добром путу да до њега дођу. Почело је са класичним приступом и алгоритмима заснованим на машинском учењу, временом су све већу улогу у детекцији људске позе добиле конволуционе неуронске мреже и оне данас представљају основу за детекцију позе тела. У почетку је било потребно неколико камера различитог типа, које снимају из другачијих углова, како би се одредили потребни параметри у 3D простору, док се сада са било ког паметног телефона у реалном времену може снимати и добијати детектована људска поза у 3D простору. Раније су само одређени програмери имали могућност изучавања детекције људске позе, због хардверских препрека, где су били потребни рачунари са напредним перформансама како би се иоле могла покренутти израчунавања, док је данас на скоро свим рачунарима могуће вршити истраживање и рад на побољшању алгоритама за детекцију позе. Велике компаније, као што је “Google”, су показале интересовање за ову грану рачунарског вида и раде на побољшању технологија, тако да је јасно да ће у будућности детекција људске позе бити све коришћенија и да програмери овај проблем неће избегавати у својим апликацијама, такође ће бити простора за нове идеје примене ове технологије.

Најпопуларнија библиотека за детекцију позе тела је свакако **OpenPose**. Метод који користи је веома брз и ефикасан, пре свега због могућности паралелизације извршавања алгоритма уз помоћ CUDA-е. Као улаз се користи RGB слика, уз помоћ мапа поверења и поља афинитета делова одређују се кључне тачке на телу и оријентација делова тела у односу на друге. Пошто је представник bottom-up приступа, један од проблема јесте када се повезују делови тела на сликама на којима постоји више особа, за ту сврху се користи greedy bipartite matching алгоритам. Овај метод се унапређује из године у годину и представља тренутно најперспективнији метод за детекцију позе тела. Главни недостатак јесте то што је за брзе и квалитетне резултате потребно имати CUDA-у која је подржана само од стране одређених NVIDIA графичких картица.

У раду је обрађено и истражено још пар библиотека за детекцију позе тела, **AlphaPose** која је представник top-down приступа и ради на принципу препознавања особа на слици, које се прво уоквирују такозваним „оквирним кутијама“, што је и проблем овог приступа, јер се појављују додатне непотребне оквирне кутије и особа може бити лоше уоквирена, имплементирани су помоћни алгоритми (STN, SPPE, SDTN), како би се решио овај проблем, што се успешно одразило и време обраде фотографија је минимизовано, иако је овај метод намењен препознавању више особа на слици, са повећањем особа на слици смањује се брзина обраде фотографије. **Detectron** је још један моћан представник библиотека за препознавање позе тела, он користи Mask R-CNN алгоритам, ради тако што се кључне тачке на телу особе препознају као “врући делови” тачније за сваку кључну тачку на телу се креира маска где је јасно обележен део тела за који је та маска задужена. Затим се ти делови спајају и добија се потпуна слика скелета.

До сада не тако популаран framework **MediaPipe** се показао као одличан представник технологија за препознавање позе човека, из простог разлога, јер ради у реалном времену и могуће га је користити на свим паметним уређајима без инсталације додатних програма или хардверских ограничења, ова технологија представља будућност препознавања објекта и иза ње стоји озбиљна компанија (Google) која улаже огромна средства у унапређење овог framework-а. За детекцију људске позе користи BlazePose метод који је представник regression-based приступа и он је „lightweight“ CNN за детекцију позе човека. Ово га чини погодним за праћење фитнес покрета и препознавање знаковног језика. За препознавање позе тела човека користи топлотне мапе и регресију истовремено.

У овом раду је представљена демо апликација за препознавање сигнала фудбалског судије помоћника. Апликација је развијена у програмском језику Python, уз помоћ PyCharm окружења и библиотека OpenCV, Tkinter и MediaPipe. Tkinter библиотека је коришћена за реализацију корисничког интерфејса, OpenCV је коришћен као помоћна библиотека за обраду фрејмова, док је MediaPipe коришћен за препознавање позе тела, захваљујући томе што се показао као најпогоднија библиотека за ову врсту апликације. Апликација имплементира препознавање сигнала фудбалског судије помоћника на основу изабраних видео записа и на излазу даје резултат о препознатим сигналима, такође води евиденцију о историји тренирања, замишљено је да се користи као помоћно средство судијама помоћницима при вежбању показивања сигнала.

# Литература

- [1] FIFA, „FIFA,“ GLT, 01. 6. 2016.. [На мрежи]. Available: <https://www.fifa.com/technical/football-technology/standards/goal-line-technology>. [Последњи приступ 20. 09. 2022.].
- [2] FIFA, „FIFA,“ VAR, 04. 05. 2018.. [На мрежи]. Available: <https://www.fifa.com/technical/football-technology/standards/video-assistant-referee>. [Последњи приступ 20. 09. 2022. ].
- [3] vj, „A Comprehensive Guide to Human Pose Estimation,“ [На мрежи]. Available: <https://www.v7labs.com/blog/human-pose-estimation-guide>.
- [4] C. S. Alexander Toshev, „DeepPose: Human Pose Estimation via Deep Neural Networks,“ ArXiv, 17. 12. 2013.. [На мрежи]. Available: <https://arxiv.org/abs/1312.4659>. [Последњи приступ 22. 08. 2022.].
- [5] G. H. T. S. S.-E. W. Y. S. Zhe Cao, „RMPE: Regional Multi-person Pose Estimation,“ Cornell University, 18. 12. 2018.. [На мрежи]. Available: <https://arxiv.org/abs/1812.08008>. [Последњи приступ 02. 09. 2022.].
- [6] S. X. Y.-W. T. C. L. Hao-Shu Fang, „RMPE: Regional Multi-person Pose Estimation,“ ArXiv, 01. 12. 2016.. [На мрежи]. Available: <https://arxiv.org/abs/1612.00137>. [Последњи приступ 06. 09. 2022.].
- [7] G. G. P. D. R. G. Kaiming He, „Mask R-CNN,“ ArXiv, 20. 03. 2017.. [На мрежи]. Available: <https://arxiv.org/abs/1703.06870>. [Последњи приступ 08. 09. 2022.].
- [8] J. T. H. N. C. M. E. U. M. H. F. Z. C.-L. C. M. G. Y. J. L. W.-T. C. W. H. M. G. M. G. Camillo Lugařevi, „MediaPipe: A Framework for Building Perception Pipelines,“ ArXiv, 14. 06. 2019.. [На мрежи]. Available: <https://arxiv.org/abs/1906.08172>. [Последњи приступ 09. 09. 2022.].
- [9] Google, „Live ML anywhere,“ Google, 01. 02. 2021.. [На мрежи]. Available: <https://google.github.io/mediapipe/>. [Последњи приступ 09. 09. 2022.].
- [10] K. Y. a. J. D. Alejandro Newell, „Stacked hourglass networks for human pose estimation.In European conference on computer vision,“ Спрингер, 2016., p. 483'499.
- [11] P. Radzki, „Detection of human body landmarks - MediaPipe and OpenPose comparison,“ HearAI, 03. 04. 2022.. [На мрежи]. Available: <https://www.hearai.pl/post/14-openpose/>. [Последњи приступ 11. 09. 2022.].
- [12] FSS, „Pravila fudbalske igre,“ Fudbalski savez Srbije, 01. 08. 2022.. [На мрежи]. Available: <https://fss.rs/pravila-igre-2022-23/>. [Последњи приступ 12. 09. 2022.].
- [13] Google, „MediPipe Pose,“ Google, 02. 01. 2022.. [На мрежи]. Available: <https://google.github.io/mediapipe/solutions/pose.html>. [Последњи приступ 13. 09. 2022.].