

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR

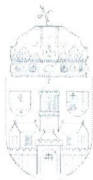


SZAKDOLGOZAT

**OE-NIK
2025**

Hallgató neve:
Hallgató törzskönyvi száma:

**Által Ádám
T/006680/FI12904/N**



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2025. 05. 20.

.....
hallgató aláírása

Tartalomjegyzék

1. Bevezetés	2
2. Irodalomkutatás	4
2.1. Hasonló megoldások	4
2.2. A játékosok felismerése	7
2.3. A labda felismerése	9
3. Rendszerterv	11
3.1. Specifikáció	11
3.2. Programnyelv	11
3.3. A képek feldolgozása	11
3.4. A játékosok és a labda felismerése	12
3.5. Az objektumok klasszifikációja	13
3.6. A les megállapítása	13
3.7. Felhasznált bemenetek	14
3.8. Kimenet	14
3.9. Komponens diagram	15
Ábrajegyzék	16
Táblázatok jegyzéke	17
Irodalomjegyzék	18

1. fejezet

Bevezetés

Modern világunkban napról napra egyre nagyobb szerepe jut a sportinformatikának. Ez a folyamat megfigyelhető a labdarúgás területén is (gondoljunk csak a videóbíró pár éve történt bevezetésére), azonban ennek a szakterületnek még bőven van hova fejlődni. A bírói döntések közül az egyik legmegosztóbb a leshelyzetek meghatározása, melyet ma már a videóbíró is segít, azonban a játékosok pozícióját kézzel rajzolják be, ezáltal fennáll a hiba lehetősége. Dolgozatom célja ezen "emberi tényező" eltávolítása lesz a játékosok pozíciójának, és ezáltal a leshelyzet meghatározásának megállapításával.

De hogy is van pontosan ez a lesszabály? Az IFAB¹ hivatalos szabálykönyve[1] alapján 2021 szeptemberében a lesszabály a következőként fogalmazható meg: a játékos leshelyzetben van, ha a labda elrúgásának pillanatában:

- a labdánál közelebb tartózkodik a gólvonalhoz
- a fej, test vagy láb bármely része az ellenfél térfelén van és
- a fej, test vagy láb bármely része közelebb van az ellenfél gólvonalához, mint az ahhoz második legközelebbi játékos (itt illik megjegyezni, hogy az első legközelebbi játékos legtöbbször a kapus)

A játékosok kezeit és karjait nem vesszük figyelembe a les megállapításánál, illetve nincs lesen egy játékos, ha egy vonalban van az utolsó előtti, vagy a két utolsó játékosal. Különbséget kell tenni aktív és passzív les között is: akkor tartózkodik aktív lesen egy játékos, ha labdához ér, vagy valamilyen formában zavarja az ellenfél játékosait.

A fentiek alapján tehát egy olyan szoftver elkészítése a cél, ami egyrészt egy képről meg tudja állapítani az azon szereplő játékosok pozícióját, másrészt a pozíciók alapján el tudja dönteni azt, hogy ki az aki leshelyzetben tartózkodik. Ehhez a következőkben megvizsgálók néhány olyan, már létező képfeldolgozást illetve neurális hálót felhasználó megvalósítást(2. fejezet2), ami képes lehet felismerni és külön választani a bevitelről számomra szükséges területeket, illetve egy olyan algoritmust, mellyel a már jól behatárolt labdarúgók pozícióiból megállapítható hogy ki tartózkodik leshelyzetben, majd ezek

¹International Football Association Board, a Nemzetközi Labdarúgó-szövetség (FIFA) Szabályalkotó Testülete

közül kiválasztom és részletesebben leírom(3. fejezet3) az általam leghatékonyabbnak véltet. A célom a Szakdolgozat tárgy végére, hogy megvizsgáljam, a már létező drága, sokkamerás megvalósításokkal szemben van-e alapja egy egy kamerát használó, kis anyagi forrásokkal dolgozó projektnek, amit akár alacsonyabb osztályokban is tudnának használni.

2. fejezet

Irodalomkutatás

2.1. Hasonló megoldások

A következőkben áttekintem néhány, a témával foglalkozó projekt megvalósítását, módszereit. Ezt követően megvizsgálom a feladat megvalósítása során felmerülő problémákat, kihívásokat, ezekre megpróbálok megoldást vagy alternatívát találni, majd kiválasztom az általam felhasználandó módszereket.

2021 nyarán a FIFA a honlapján beszámolt arról, hogy videokonferenciát tartott a korábban általuk, a technológiai fejlesztések céljából létrehozott Working Group of Innovation Excellence-szel (magyarul nagyjából Innovációs Munkacsoportnak lehetne hívni), mely során bejelentették, hogy a videóbíró bevezetésének sikere után szeretnék továbbfejleszteni a rendszert egy félautomata lefelismerő funkcióval. Ennek célja elsősorban nem a leshelyzet megállapítása lenne, hanem egy eszköz biztosítása a játékvezetők számára, mely segítségével könnyebben meghozhatják a döntésüket - a már használatban lévő gólvonal technológiához hasonló módon. Ezen konferencia során a felmerülő kihívások között felmerült a labda elrúgásának pillanata (annak pontos megállapítása), illetve a testrészek helyes felismerése, ugyanis még kihívást jelent a játékosok "csontvázának" modellezése.[2] Ez a rendszer végül a 2022-es világbajnokságtól kezdődően egyre több helyen bevezetésre kerül(t). A szóban forgó katari világbajnokságon a stadionok tetőszerkezetére erősített 12 kamerával és egy a labdába helyezett szenzor segítségével működött. A pontos implementációról nem esett említés. [3] Ez egy rendkívül fejlett, nagy költségekkel járó megoldás, ezen dolgozat keretein belül természetesen nem áll rendelkezésemre ilyen jellegű technológia, ennek megfelelően nyílt forráskódú eszközökre és korlátozott számú, online fellelhető videóforrásra támaszkodom. Célom egy hasonló funkcionalitású, de jóval egyszerűbb rendszer létrehozására.

2009-ben egy olasz csapat írt egy elég átfogó cikket a témában.[4] Ebben egy valós idejű lesdetektáló rendszer megvalósíthatóságát vizsgálták. Három-három nagy felbontású kamerát helyeztek el mindkét oldalvonal mentén úgy, hogy az optikai tengelyek a gólvonallal párhuzamosak legyenek a perspektívahibák elkerülése érdekében. Ezek felvételeit szinkronizálták, és párhuzamosan feldolgozták, működésüket pedig egy hetedik, központi "felügyelő" csomópont szinkronizálta újra, illetve meghozta a végső döntést. A

felvételek feldolgozásának első lépése a háttér eltávolítása volt. Mivel mozgóképpel dolgoztak, ehhez egy mozgásérzékelő algoritmust használtak, amely a legalább 2,5 másodpercig mozdulatlan pontokat feltételezte háttérnek. Ezek után geometriai módszerekkel kivonták a játékosok árnyékait is. A szegmentáció után szükség volt arra, hogy minden elkülönített objektumhoz hozzárendeljék hogy hova tartozik, mindezt emberi beavatkozás nélkül. Ehhez először egy clusterező algoritmussal és véletlenszerűen választott bemenetekkel csoportokat hoztak létre, majd futásidőben minden szegmentált objektumot hozzárendeltek a létrehozott clusterek egyikéhez. Ezen klasszifikáció után a játékosokat megfeleltették egy "határdobozzal", ami segítségével követni tudták a mozgásukat. A labda felismeréséhez is a mozgó objektumokat használták, azonban először méret alapján kiszűrték a lehetséges pozícióit (a mérkőzés során később, amikor már rendelkezésre állnak előzetes információk azokat felhasználva), majd egy mintafelismerő algoritmussal kiválasztották ezek közül a labda tényleges helyzetét. Több egymást követő kép segítségével a labda sebesség- és mozgásának irányváltozásait megvizsgálva a lövés pillanatát is meg tudták állapítani. Ezzel minden rendelkezésre áll a leshelyzet megállapításának döntéséhez. A fent írtak alapján tehát a csomópontok által feldolgozott adatok egy központi csomóponton kerülnek szinkronizálásra, és itt fog megszületni az eredmény is. A szinkronizáció során a felügyelő csomópont egy virtuális pálya segítségével egyesíti a kapott adatokat, a nagyon eltérő bemeneteket átugorva, vagy a megbízhatóbbnak tűnőeket validálva. A rendszer egyaránt tud azonnal jelezni leshelyzet esetén - ekkor csak azon képek kerülnek vizsgálatra, amelyeken hirtelen változás történt a labda mozgásában; illetve meg tudja különböztetni hogy aktív vagy passzív leshelyzetről van szó, egy 3 másodperces ablak során megvizsgálva hogy a lesen lévő játékos játékba tud(na)-e avatkozni.

Egy 2013-as tanulmányban a teheráni Payame Noor Egyetem két munkatársa is foglalkozott a témával.[5] Ebben különösen nagy hangsúlyt fektettek a kamerafelvétel perspektivikus hibáinak kiküszöbölésére. A háttér felismerését Kálmán-szűrő segítségével végezték el, majd felismerték az objektumokat, elkülönítve a játékosokat, játékvezetőt és a labdát. Egy algoritmussal felismerték és megjelölték a leshelyzetben lévő játékosokat, majd megvizsgálva a térfelet, a passzot játszó játékost, és a passz irányát megállapították hogy lesen van-e az adott játékos.

Egy másik, Indiában elkészített megvalósítás a játékosok kontúrjainak a felismerésével oldotta meg a feladatot.[6] A megoldás során a csapatok mezeinek színének előzetes ismeretével éldetektálást hajtanak végre a képen, Gauss-szűrővel csökkentve a zajos/hibás élek számát. Ezt követően a játékosok alapvonalhoz legközelebbi pontjait összehasonlítva megszületik a végső döntés.

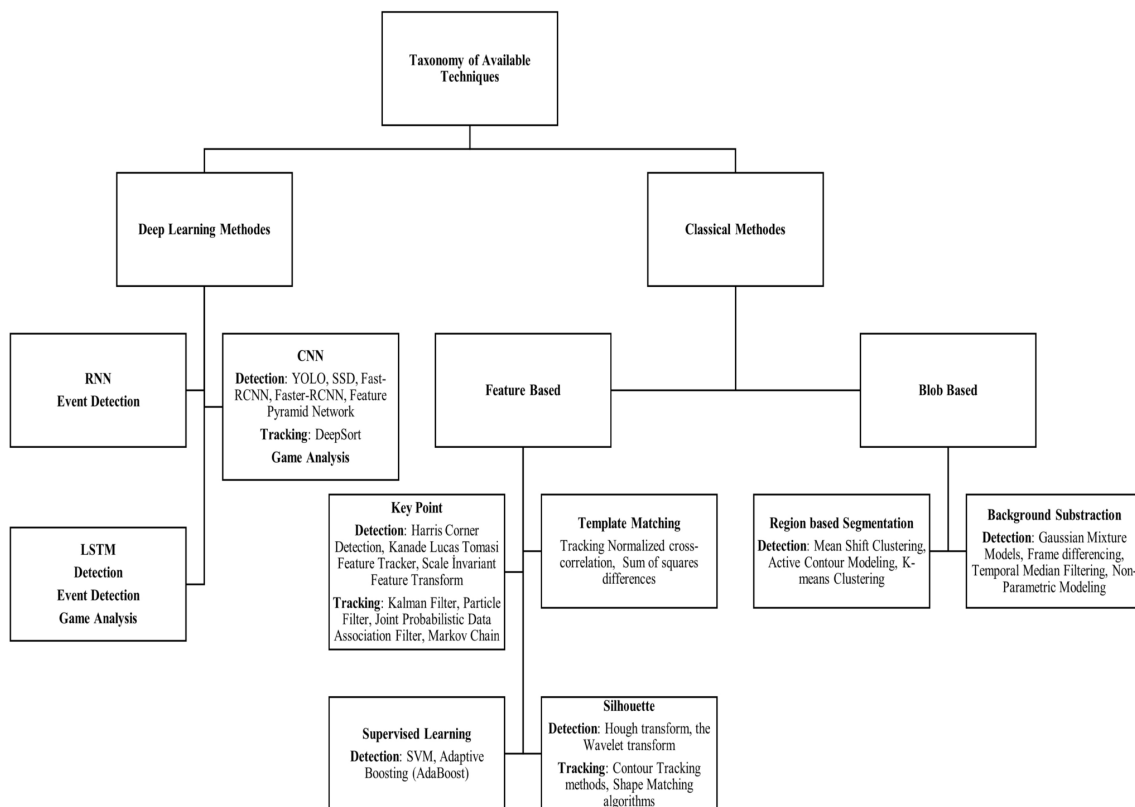
2025 februárjában Rochesterben született a képfeldolgozós megoldások közül egy egyszerű, egykamerás megoldás a valós idejű lesfelismeréshez. [7] Ennek célja a drága, sokkamerás, érzékelős megvalósításokkal szemben egy megfizethető megoldás megtalálása volt, melyet alacsonyabb szinteken, félprofi, vagy akár amatőr csapatok is tudnának használni. Ebben a megoldásban először a játéktér szegmentálták, ezután megkeresték a távlatpontot, aminek a segítségével kiküszöbölhetőek a 3D-s tér 2D-ben való ábrázolásának hibái. Végezetül a játékosok "kulcspontjait" (térdek, vállak, lábak) kereste meg,

hogy ezekkel húzza be a lesvonalat.

Ugyan szigorúan véve nem lesfelismerésről szól, de számomra megkerülhetetlennek tűnt egy 2022 év végén publikált cikk. [8] Ebben az elmúlt évek labdarúgással kapcsolatos kutatásait mutatják be, egyaránt deep learning és általuk "klasszikus"-nak nevezett módszerek felhasználásával. Kitértek ezen kutatások jelenlegi kihívásaira, illetve 4 csoportba sorolták a labdarúgással kapcsolatos kutatásokat:

- a játékosok detektálása és követése;
- a labda detektálása és követése;
- esemény detektálás (ide sorolják a lesfelismerést is);
- játék analízis.

Ezt követően megvizsgálták az általuk elérhetőnek tartott megoldási lehetőségeket (2.1). A klasszikus metodológiákat két részre osztották, 'blob' illetve 'feature' alapú megoldásokra. Az előbbiek közé vették a terület alapú szegmentációt (pl. k-means) és a háttér kivonást. Ezek egyaránt mozgó, azonos színű vagy textúrájú részleteket tekintenek játékosnak és labdának. Ezek a módszerek a cikk szerint egész jól működnek játékosok felismerésére, azonban a labda kis mérete, illetve játékosok általi részleges vagy akár teljes takarása miatt problémák merülhetnek fel. A 'feature' alapú módszereknek négy fajtáját állapították meg: a kulcspont keresés (pl. Kálmán filter vagy KLT tracker) során, ahogy a név is sugallja, az algoritmus meghatározza bizonyos kulcspontjait az objektum(ok)nak, és ezeknek az egymást követő frameken való viszonyuk segítségével követi az adott testet; a sziluett követés (pl. Hough transzformálás vagy kontúr követés) során egy bizonyos modellhez hasonló szilvetteket vagy kontúrokat keres a program, majd a megtalálás utáni képeken becsléssel követi az adott objektum lehetséges helyzetét; a felügyelt tanulás (pl. AdaBoost) betanítás után a megtanult tulajdonságok alapján keres hasonló testeket; a template matching ('sablon egyezés', pl. normalizált kereszt-korreláció) forgatás vagy más transzformáció által okozott változásokat keres. Ezen módszerek közös tulajdonsága, hogy egyaránt az objektumok megjelenése alapján detektálnak és követnek, azonban nem kifejezetten alkalmasak valós idejű megoldásokhoz, és csak specifikus problémák oldhatóak meg velük. A gépi tanulási metodológiáknak három fajtáját nevezték meg: konvolúciós neurális háló (CNN), ismétlődő neurális háló (RNN), hosszú rövidtávú memória (LSTM). Mindhárom megoldásról hoznak példákat, melyek alapján azt állapították meg, hogy a konvolúciós neurális hálók 2D-s változata nem képes időbeli információkat kivonni, így ajánlott a 3D-s módszert alkalmazni, mely képes egyaránt idő- és térbeli információkat kivonni. Az ismétlődő neurális hálók képesek ugyan jelenlegi és korábbi megfigyelések alapján is az időbeli kivonásra, azonban ezen architektúrák csak rövidtávú memóriával rendelkeznek, így valós idejű szituációkban nem reális a használatuk. Erre megoldást nyújthat a hosszú rövidtávú memória, amely egy saját memória egységgel rendelkezik, ami meg tudja állapítani, bizonyos rejtett állapotok megjegyzésére mikor van szükség és mikor nincs.



2.1. ábra. A [8] cikk által felvázolt megoldások

2.2. A játékosok felismerése

Az első és talán legfontosabb kérdés maguknak a játékosoknak a felismerése. Ezt hagyományos képfeldolgozást használó alkalmazásoknál strukturálisan három részre lehet osztani: szegmentálás, követés és klasszifikáció. Ezek mind kulcsszerepet játszanak egy működő algoritmus felépítésében.

A mozgó objektumok detektálásának három hagyományos módszere van: *temporal differencing*, háttér kivonás (*background subtraction*), optikai áramlás(*optical flow*).[9]

	háttér kivonás	temporal differencing	optical flow
előnyök	relatív statikus hátterekhez kiváló	alkalmazkodik a dinamikus hátterekhez; könnyű implementálni	hatékony dinamikus háttereknel
hátrányok	különböző statisztikai modellezési feltevések közül kell választani	rosszul működik egész mozgó objektumok esetén; nagyon zajérzékeny	Számításlag összetett; erősen zajérzékeny
komplexitás	közepes	egyszerű	bonyolult

2.1. táblázat. A mozgásfelismerő módszerek összehasonlítása[10]



2.2. ábra. Kék színű pont jelzi a támadó és a védő legközelebbi pontját

A klasszifikáció is többféle módon történhet. Kézenfekvő megoldásnak tűnik a játékosok mezeinek színét felhasználva elkülöníteni őket. Felmerül azonban az a kérdés, hogy a vizsgált bemenet elején manuálisan állítsuk-e be, hogy melyik szín melyik csapatot reprezentálja, vagy egy felügyeletet nem igénylő (*unsupervised*) algoritmust használunk, mely automatikusan állapítja meg a felismert játékosok színét. Amennyiben utóbbi mellett döntünk, a [11]-ban háromféle klaszterező algoritmus merül fel: MBSAS (*Modified Basic Sequential Algorithmic Scheme*, módosított alapvető szekvenciális algoritmikus séma): melynek során az objektumokat reprezentáló vektorokat hozzárendeljük egy már létező klaszterhez, vagy egy újat hozunk létre erre a célra, mindezt a már létrehozott klaszterektől való távolságtól függően; BCLS (*Balanced Clustering with Least Square regression*, kiegyensúlyozott klaszterezés legkisebb négyzetek módszerével): ez egy kompetitív algoritmus, amelyben a csoportokat véletlenszerűen inicializáljuk, majd valamilyen összehasonlítás alapján kiválasztjuk a győztes klasztert, melyet frissítünk, végzetül pedig hozzárendeljük az objektumokat a klaszterekhez; Isodata (vagy k-means): a csoportokat szintén véletlenszerűen inicializáljuk, majd addig prezentáljuk a vektorokat, amíg a klaszterek változnak, minden egyes prezentálás során a klasztereket a bemutatott vektorokkal való eltérés függvényében frissítjük.

Ezekkel szemben használhatunk a játékosok felismeréséhez neurális hálón alapuló megoldást is. A [12]-ban a kis méretű játékosok detektálásával foglalkoztak. A megoldás során egy önfelügyelt (*self-supervised*) rendszert hoztak létre, ami felcímkézetlen (*unlabeled*) adatokkal is tud dolgozni, ami a viszonylag kevés jó minőségű adathalmaz miatt előnyös lehet. Először felismerték a játékosokat, majd ennek eredményeinek felhasználásával követték őket. A felismerés három lépésben zajlik: először elvégeznek egy kezdeti automatikus felcímkézést és javítják a fals pozitívak elnyomásával és fals negatívak hozzáadásával. Ehhez pálya- és vonaldetektálást, illetve blob detektálást alkalmaznak. A következő lépésben ezen a javított adathalmazon végeznek finomhangolásokat. Majd végül a tanító hálózatot áttöltik egy kisebb hálózatba, ami a végleges detektáló lesz. A követés során előbb egy spatial consistency association-t (térbeli konzisztencia összekapcsolás) végeznek, majd ennek a hibái kiküszöbölésére újraazonosítást és vizuális konzisztencia

mérést végeznek. A [13] forrásban a televíziós közvetítések során történő játékosdetektálást vizsgálták meg. Ennek során egy valós idejű, teljes mértékben automatizált rendszert hoztak létre. Ebben egy saját, lightweight CNN architektúrát alkalmaznak, ami a YOLO-n alapul, de sportközvetítésekre van optimalizálva. A rendszer az előző forráshoz hasonlóan adatasszociáció segítségével végez utómunkát.

A [14]-ban kizárólag a játékosok detektálásával foglalkoznak. Ehhez egy lightweight, kompakt CNN-t használnak, ami egy javaslatkészítő (proposal) és egy klasszifikációs networkból áll. Ezek során a program képkockarészeket generál a teljes képből, majd azokat osztályozza aszerint, hogy az adott részlet játékos, vagy nem. Ha játékos, megállapítja annak csapatát is. Ezen rendszer előnye, hogy egy rendkívül kis méretű, kevés paraméterrel rendelkező, de viszonylag gyors megoldás született.

Egy 2024 júniusában publikált kutatás[15] során a YOLOv5 rendszer egy továbbfejlesztett változatát használták valós idejű játékosdetektáláshoz. A rendszer először egy YOLOv5 algoritmussal elvégzi az előzetes detektálást, majd egy SimSPPF (Simplified Spatial Pyramid Pooling) modullal segíti a program különböző méretű objektumok felismerését. Ezt követően egy GhostNet modullal csökkentik a számítási komplexitást, egy slim scale detekciós réteggel megjósolja az objektumok bounding boxait és azok konfidenciaértékét. Végül a kimenet létrejötte előtt egy nem-maximális elnyomással megszüntetik az átfedő vagy redundáns bounding boxokat.

2.3. A labda felismerése

Leshelyzet megállapításához kulcsfontosságú tudni a labda pontos pozícióját is. A labdafelismerés egy elég gyakran vizsgált feladat, számtalan ilyen projekt született az évek során, ezek hagyományosan képfeldolgozós változatait két kategóriába lehet sorolni: közvetlen (direkt) vagy közvetett (indirekt) módszerek.[16]

A közvetett módszerek valamilyen, a labda mozgásával kapcsolatos előzetes tudásra alapozva különböztetik meg a labdát a többi objektumtól. A [17]-ban a labda pályájának vizsgálatával állapítják meg annak pozícióját több "labda-szerű" jelölt közül. A [18]-ban a labda pályáját síkgörbéként modellezik. Ezeket a síkokat görbületi küszöböléssel és 3D vonalaktól vett legkisebb távolság segítségével határozzák meg. Ezek a módszerek jól alkalmazhatóak videóindexeléshez, azonban valós idejű események észlelésére kevésbé, ugyanis nem képesek a labda helyzetét minden képkockán kiértékelni.[16]

A közvetlen módszerek ezzel ellentétben minden képkockán képesek megállapítani a labda helyzetét annak formája, mérete illetve színe alapján. A [19]-ban *Support Vector Machine* (SVM) módszerrel klasszifikálták a labdát. Ez jól látható labda esetén igen jól működött (98,3% detektálási, 0,2% fals pozitív arány), azonban a nehezen látható labdák esetén ez jelentősen gyengébb volt (76,2% detektálási, 2,6% fals pozitív arány). A [20]-ban hullám dekompozíció és független komponensanalízis (*Independent Component Analysis*, ICA) segítségével a nehezen látható labda esetén jobb eredményeket értek el (92,08% és 87,06% detektálási arány). A [16]-ban először kör Hough transzformációt alkalmaznak a mozgó objektumokra a képen, hogy megtalálják a leginkább kör alakút.

Ezután egy skálainvariáns jellemző transzformáció (*Scale-invariant feature transformation*, SIFT) használatával megvizsgálják, hogy a felismert objektum valóban labda-e. Ezen módszerrel leegyszerűsíthető a betanítás folyamata. Az eredmények detektálás szempontjából a két másik vizsgált módszer között (83.08-90.3%) helyezkedett el, azonban nagyon nagy volt a fals pozitív eredmények előfordulása (9.7 és 16.91%).

A labdafelismerésre is születtek neurális hálót használó megoldások. A [21]-ban több neurális hálós megoldás teljesítményét hasonlítja össze labdadetektálásra korlátozott hardveres helyzetekben. A cikkben megvizsgálták a MobileNETV2-t és V3-at, a YOLO és TinyYOLO v3-at és v4-et. Az eredmények alapján megállapítható, hogy a MobileNet verziók jobb teljesítményt nyújtanak CPU-n, mint a GPU-ra optimalizált YOLO implementációk.

A [22]-ban gépi tanulással detektálták a labdákat. Ez a program először egy kétlépcsős puffer mediánszűrős háttérmodellezéssel azonosít olyan blobokat, amelyek mozgó objektumokat tartalmaznak. Ezt követően egy deep learning alapú megoldással osztályozzák ezeket a képrészleteket három kategóriába: labda, játékos, háttér. Ehhez egy előtanított neurális hálót használnak, amelyet transzfer tanulással tanítanak be.

3. fejezet

Rendszerterv

3.1. Specifikáció

Az irodalomkutatás alapján kiderült, hogy számos megvalósítás született lesfelismerésre, játékos- és labdadetektálásra az elmúlt évek során. Azonban ezek szinte teljes mértékben többkamerás, erősen hardverigényes megvalósítások, melyek használata nem reális egy jelentős pénzügyi korlátokkal operáló, alacsonyabb vonalban lévő klub számára. Dolgozatomban meg fogom vizsgálni, hogy egy egykamerás, úgymond "házi készítésű" rendszer milyen eredményeket képes produkálni egy drága, hivatalos megvalósítással szemben, illetve, hogy ez reális megoldást jelenthet-e az alacsonyabb osztályokban.

3.2. Programnyelv

Egy informatikai projekt talán legfontosabb döntése, hogy milyen programozási nyelven fogjuk írni programunkat. Ebben a projektben a Python open-source programozási nyelvet fogom használni annak nagy népszerűségéből fakadó támogatottsága és jó modularitása (rengeteg használható könyvtár) miatt. Amennyiben szükséges, a kódot helyenként CUDA C nyelv használatával lehetne gyorsítani.

3.3. A képek feldolgozása

Bemenetként a szoftver egy videót fog kapni, melyet be kell olvasnia, és feldolgozni a további műveletek előkészítése céljából. Erre a feladatra a python számos ingyenesen hozzáférhető könyvtárat biztosít, melyek közül az OpenCV-t és a Scikit-Image-et is hasonlóan alkalmasnak találtam a projekt számára, így végül az döntött az OpenCV mellett, hogy már dolgoztam vele korábban.

	Pillow	OpenCV	Scikit-Image	PyTorch
előnyök	sok fájlformátumot támogat, egyszerű a használata	függvények széles köre érhető el, jó teljesítmény	jól dokumentált, jól integrálható tudományos számításokkal	jól használható deep learninggel kapcsolatos feladatokra, gyors
hátrányok	lassabb a többinél, videófeldolgozásra nem optimális	valamivel komplexebb	bizonyos feladatokra nincs függvénye	általános képfeldolgozásra kevésbé alkalmas

3.1. táblázat. A python néhány népszerűbb képfeldolgozó könyvtárának összehasonlítása

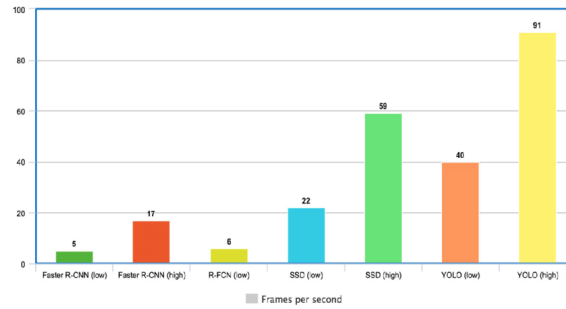
3.4. A játékosok és a labda felismerése

Ahogy az irodalomkutatásban látható, számos kizárólagosan képfeldolgozás alapú megoldás született már a lesfelismerés problémájára, azonban ezek javarészt régi, 10-15 éves megvalósítások. Manapság már egyre kevésbé jellemzőek az ilyen módszerek az informatikában, és bár játékos- és labdadetektálási feladatra jó számmal találunk gépi tanulással megoldást, a lesfelismerésben ez nem mondható el. Emiatt dolgozatomban is gépi tanulást fogok használni.

	Faster R-CNN	Mask R-CNN	YOLO	SSD
előnyök	az egyik legpontosabb modell	szegmentálásnál precízen állapítja meg az objektumok helyzetét	jól képes valós idejű feldolgozásra	valamivel gyorsabb, mint a Faster R-CNN
hátrányok	nem képes valós időben feldolgozni	lassabb, mint a Faster R-CNN	kisebb objektumokat nehezebben ismer fel	kisebb pontosságú a többi modellhez képest

3.2. táblázat. Néhány népszerűbb objektumdetektálási módszer összehasonlítása [23]

Az objektumdetektálás feladatára többféle gépi tanulási módszer is létezik, azonban mivel ebben a dolgozatban a célom egy valós idejű rendszer létrehozása, a legtöbb megoldás nem éri el a kellő feldolgozási sebességet ezért limitált választási lehetőség áll rendelkezésre. Ahogy a mellékelt ábra 3.1 is mutatja, a YOLO messze a leggyorsabb a módszerek között, és pontosságban is az élmezőnyben jár. Emellett a YOLO egy viszonylag egyszerűen megvalósítható, jelentős online támogatói és erőforrás-rendszerrel rendelkező megoldás. Mindezek tudatában az én döntésem is a YOLO mellett tettem le, amelynek az épp legújabb verzióját fogom használni, ez az írás pillanatában (2025 május) a YOLOv12, azonban ennek még jelenleg nincsen szegmentálásra is optimalizált változata,



3.1. ábra. Néhány módszer feldolgozási sebességének összehasonlítása alacsony és magas felbontás esetén [23]

így kezdetben a 11-es verziót fogom használni, de amennyiben készülnek újabb verziók a fejlesztés során, ez a jövőben még változhat. Mivel a YOLO működése során bounding boxokat jósol meg, ezért elképzelhető, hogy egy bizonyos objektumra több ilyen boxot is fog találni. Ennek kiküszöbölése érdekében egy nem-maximum elnyomással kiválasztjuk a legmegfelelőbbeket. [15]

3.5. Az objektumok klasszifikációja

A YOLO ugyan végez egy alapszintű osztályozást, tehát meg tudja különböztetni például a játékosokat a labdától, azonban nem képes ettől részletesebb klasszifikációra, ezért ehhez egy plusz lépésre van szükségünk. Mivel a projekt célja a les megállapításának automatizálása lenne, ezért a klasszifikációt is érdemes lenne emberi beavatkozás szükségesége nélkül elvégezni, így egy nem felügyelt klaszterezési módszert kell alkalmaznunk. Erre a [11]-ben végeztek tesztek, amelyben alapján a k-means klaszterezés módszere tűnik a legmegbízhatóbbnak. A k-means gyors, így alkalmazható valós időben, jól teljesít RGB színtérben, és teljes mértékben nem felügyelt, így nem igényel emberi beavatkozást. Emellett azért érdemes azt is megemlíteni, hogy a hasonló színű mezek a két csapat vagy a játékvezető között okozhatnak problémákat. Mindezek tudatában én is ezt a módszert fogom alkalmazni.

3.6. A les megállapítása

A program utolsó lépése a leshelyzet tényének megállapítása lesz. Ez elsőre egyszerű feladatnak tűnik, azonban nehéz olyan megoldást találni, ami egyaránt képes valós időben, és külső emberi beavatkozás nélkül működni. Az egyik ilyen lehetséges megvalósítás során homográfiai becsléssel a pálya ismert vonalai mentén leképezhető egy homográfia mátrix, majd ebből egy két dimenziós pályakép. [4] Ezen a pályaképen már könnyen megállapítható a játékosok pontos pozíciója, behúzható a lesvonal, és meg lehet állapítani, mely támadók tartózkodnak lesen. A les szabályai között szó esik aktív, illetve passzív lesről, azonban ezen szabályozás értelmezése játékvezetőnként eltérhet, és mivel ennek a szoftvernek a célja nem a játékvezetők kiváltása, hanem segítése, ezért ezt ennek a dön-

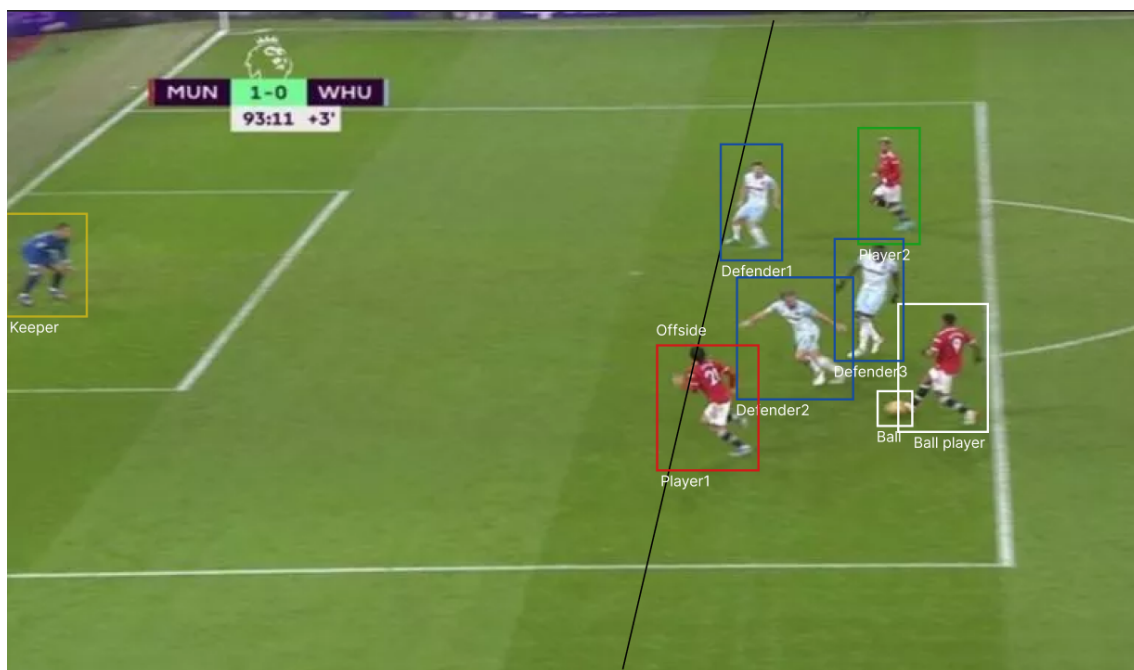
tésnek a meghozatalát a játékvezetőre bízom, ezt a szabályt nem fogom figyelembe venni a dolgozatom során.

3.7. Felhasznált bemenetek

A labdarúgásban annak népszerűsége miatt rendkívüli mennyiségű videó és adathalmaz áll rendelkezésre az online térben. Mindezek ellenére nehéz olyan bemeneteket találni, ami konkrétan a leshelyzetekre fókuszál. Az egyik kiemelkedő adathalmaz a Soccer-Net [24]. Ez az adathalmaz 550 teljes közvetített mérkőzést tartalmaz. Rendelkezésre áll a korábban többször is említett olasz kutatócsoport egy adathalmaza is.[25] Említésre méltó még a [26]-ban található adathalmaz. Ezek egyaránt hasznosak betanításra, ezért lehetőség szerint minddel ki fogom próbálni a YOLO egy betanított változatát, és megvizsgálom, melyikkel milyen eredmények érhetőek el. Ezen kívül szeretnék a projekt szelleméhez méltó alacsony osztályú tesztelési bemenetet használni, erre az angol hetedosztályú Hashtag United hivatalos Youtube csatornáján fellelhető videókat fogom felhasználni. [27] Felmerült még bennem, hogy felhasználásra kerülhetne felvétel a Football Manager nevű szimulátoros videójátékból, ugyanis a mérkőzések kamerájának állása több módon is állítható, azonban az ötletet elvetettem, mert bár a játék grafikai motorja évről évre egyre realisztikusabb, még mindig nem tudja tökéletesen reprezentálni egy valós futballmérkőzés történéseit.

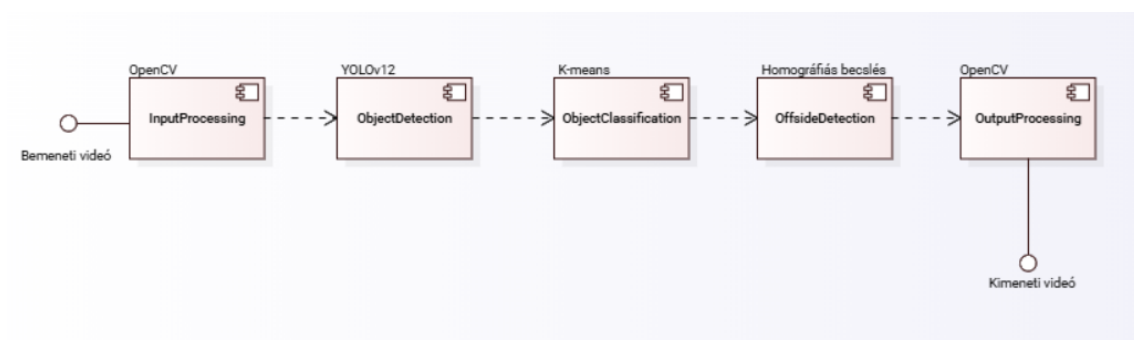
3.8. Kimenet

Mindezen módszerekkel elkészül végül a kimenet, ami egy, a lehetőségekhez mérten nagyjából valós-idejű videófelvétel lesz, melyen a játékosok *bounding box*-okkal vannak megjelölve, melyek színei és a hozzájuk társított feliratok jelölik, hogy egy adott játékos melyik csapathoz tartozik, illetve, hogy lesen tartózkodik-e éppen.



3.2. ábra. Figma-val készített vázlat a program kimenetéről. Az eredeti kép forrása:[28]

3.9. Komponens diagram



3.3. ábra. A rendszer egy előzetes terve komponens diagrammal szemléltetve.

Ábrajegyzék

2.1.	A [8] cikk által felvázolt megoldások	7
2.2.	Kék színű pont jelzi a támadó és a védő legközelebbi pontját	8
3.1.	Néhány módszer feldolgozási sebességének összehasonlítása alacsony és magas felbontás esetén [23]	13
3.2.	Figma-val készített vázlat a program kimenetéről. Az eredeti kép forrása:[28]	15
3.3.	A rendszer egy előzetes terve komponens diagrammal szemlélítve. . . .	15

Táblázatok jegyzéke

2.1. A mozgásfelismerő módszerek összehasonlítása[10]	7
3.1. A python néhány népszerűbb képfeldolgozó könyvtárának összehasonlítása	12
3.2. Néhány népszerűbb objektumdetektálási módszer összehasonlítása [23] .	12

Irodalomjegyzék

- [1] The International Football Association Board. *Laws of the Game*. 2021.
- [2] FIFA.com. *FIFA organises remote demonstration of advanced offside technology*. 2021. URL: <https://www.fifa.com/news/fifa-organises-remote-demonstration-of-advanced-offside-technology>.
- [3] FIFA.com. *Semi-automated offside technology*. 2023. URL: <https://inside.fifa.com/innovation/world-cup-2022/semi-automated-offside-technology>.
- [4] T. D’Orazio és tsai. “An Investigation Into the Feasibility of Real-Time Soccer Offside Detection From a Multiple Camera System”. *Circuits and Systems for Video Technology, IEEE Transactions on* 19 (2010. jan.), 1804 –1818. old. DOI: 10.1109/TCSVT.2009.2026817.
- [5] Mohammad Assadifard és Mehdi or Mahdi Javanmard. “Using Digital Image Processing Solutions For Offside Recognition, Instead of Assistant Referees”. *The 2013 World Congress in Computer Science Computer Engineering and Applied Computing* (2013. júl.).
- [6] Pratik N Patil, Rebecca J Salve, Karanjit R Pawar és Prof. Mrs. M. P. Atre. “Offside Detection in the Game of Football Using Contour Mapping”. 2018.
- [7] Shounak Desai. *Real Time Offside Detection using a Single Camera in Soccer*. 2025. arXiv: 2502.16030 [cs.CV]. URL: <https://arxiv.org/abs/2502.16030>.
- [8] Akan S. és Varlı S. “Use of deep learning in soccer videos analysis: survey.” (2022. dec.). URL: <https://doi.org/10.1007/s00530-022-01027-0>.
- [9] Takeo Kanade, Robert T. Collins, Alan J. Lipton és Peter J. Burt. “Advances in Cooperative Multi-Sensor Video Surveillance”. 1999.
- [10] Joonsoo Lee és Al Bovik. “CHAPTER 19 - Video Surveillance”. *The Essential Guide to Video Processing*. Szerk. Al Bovik. Boston: Academic Press, 2009, 619–651. old. ISBN: 978-0-12-374456-2. DOI: <https://doi.org/10.1016/B978-0-12-374456-2.00022-0>. URL: <https://www.sciencedirect.com/science/article/pii/B9780123744562000220>.

- [11] Pier Luigi Mazzeo, Paolo Spagnolo, Marco Leo és T. D’Orazio. “Football Players Classification in a Multi-camera Environment”. 6475. köt. 2010. dec., 143–154. old. ISBN: 978-3-642-17690-6. DOI: 10.1007/978-3-642-17691-3_14.
- [12] Samuel Hurault, Coloma Ballester és Gloria Haro. “Self-Supervised Small Soccer Player Detection and Tracking”. *Proceedings of the 3rd International Workshop on Multimedia Content Analysis in Sports*. MMSports ’20. Seattle, WA, USA: Association for Computing Machinery, 2020, 9–18. ISBN: 9781450381499. DOI: 10.1145/3422844.3423054. URL: <https://doi.org/10.1145/3422844.3423054>.
- [13] Sigurdur Sverrisson, Volodya Grancharov és Harald Pobloth. “Real-Time Tracking-by-Detection in Broadcast Sports Videos”. *Image Analysis*. Szerk. Michael Felsberg, Per-Erik Forssén, Ida-Maria Sintorn és Jonas Unger. Cham: Springer International Publishing, 2019, 399–411. old.
- [14] Keyu Lu, Jianhui Chen, James J. Little és Hangen He. “Lightweight convolutional neural networks for player detection and classification”. *Computer Vision and Image Understanding* 172 (2018), 77–87. old. ISSN: 1077-3142. DOI: <https://doi.org/10.1016/j.cviu.2018.02.008>. URL: <https://www.sciencedirect.com/science/article/pii/S1077314218300341>.
- [15] K. Zhao. “Enhancing the Performance and Accuracy in Real-Time Football and Player Detection Using Upgraded YOLOv5 Architecture.” (2024). DOI: <https://doi.org/10.1007/s44196-024-00565-x>. URL: <https://link.springer.com/article/10.1007/s44196-024-00565-x>.
- [16] Marco Leo, T. D’Orazio, Paolo Spagnolo, Pier Luigi Mazzeo és Arcangelo Distanti. “SIFT Based Ball Recognition in Soccer Images”. 5099. köt. 2008. júl., 263–272. old. ISBN: 978-3-540-69904-0. DOI: 10.1007/978-3-540-69905-7_30.
- [17] Xinguo Yu, Hon Leong, Changsheng Xu és Qi Tian. “Trajectory-Based Ball Detection and Tracking in Broadcast Soccer Video”. *Multimedia, IEEE Transactions on* 8 (2007. jan.), 1164–1178. old. DOI: 10.1109/TMM.2006.884621.
- [18] Jinchang Ren, J. Orwell, G.A. Jones és Ming Xu. “A general framework for 3D soccer ball estimation and tracking”. *2004 International Conference on Image Processing, 2004. ICIP ’04*. 3. köt. 2004, 1935–1938 Vol. 3. DOI: 10.1109/ICIP.2004.1421458.
- [19] Nicola Ancona, Grazia Cicirelli, A. Branca és A. Distanti. “Goal detection in football by using Support Vector Machines for classification”. 1. köt. 2001. febr., 611–616 vol.1. ISBN: 0-7803-7044-9. DOI: 10.1109/IJCNN.2001.939092.
- [20] Marco Leo, Tiziana D’orazio, Paolo Spagnolo és Arcangelo Distanti. “Wavelet and ICA Preprocessing for Ball Recognition in Soccer Images”. 2005.

- [21] Douglas De Rizzo Meneghetti, Thiago Pedro Donadon Homem, Jonas Henrique Renolfi de Oliveira, Isaac Jesus da Silva, Danilo Hernani Perico és Reinaldo Augusto da Costa Bianchi. “Detecting Soccer Balls with Reduced Neural Networks: A Comparison of Multiple Architectures Under Constrained Hardware Scenarios”. *Journal of Intelligent & Robotic Systems* 101.3 (2021. febr.). ISSN: 1573-0409. DOI: 10.1007/s10846-021-01336-y. URL: <http://dx.doi.org/10.1007/s10846-021-01336-y>.
- [22] P.R. Kamble, A.G. Keskar és K.M. Bhurchandi. “A deep learning ball tracking system in soccer videos”. *Opto-Electronics Review* 27.1 (2019), 58–69. old. ISSN: 1230-3402. DOI: <https://doi.org/10.1016/j.opelre.2019.02.003>. URL: <https://www.sciencedirect.com/science/article/pii/S123034021830146X>.
- [23] H Romero és Alex Morales Acosta. “A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework”. *IOP Conference Series: Materials Science and Engineering* 844 (2020. jún.), 12024. old. DOI: 10.1088/1757-899X/844/1/012024.
- [24] Adrien Delière és tsai. *SoccerNet-v2: A Dataset and Benchmarks for Holistic Understanding of Broadcast Soccer Videos*. 2021. arXiv: 2011.13367 [cs.CV]. URL: <https://arxiv.org/abs/2011.13367>.
- [25] T. D’Orazio, M. Leo, N. Mosca, P. Spagnolo és P.L. Mazzeo. “A Semi-automatic System for Ground Truth Generation of Soccer Video Sequences”. *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. 2009, 559–564. old. DOI: 10.1109/AVSS.2009.69.
- [26] Jamal Hussain Shah, Maira Afzal, Samia Riaz, Mussarat Yasmin, Seifedine Kadry és Fahad Ahmed Khokhar. “A comprehensive dataset of soccer event images for advancing automatic recognition systems”. *Data in Brief* 60 (2025), 111518. old. ISSN: 2352-3409. DOI: <https://doi.org/10.1016/j.dib.2025.111518>. URL: <https://www.sciencedirect.com/science/article/pii/S2352340925002501>.
- [27] Hashtag United. *FULL MATCH! - White Ensign vs Hashtag United*. YouTube playlist. URL: https://www.youtube.com/watch?v=3N7BkyuEBAw&ab_channel=HashtagUnited.
- [28] caughtoffside.com. (Photo) – *The potential offside moment for Marcus Rashford’s late winner for Man United vs West Ham*. URL: <https://www.caughtoffside.com/2022/01/22/photo-offside-man-united-goal/>.