NYILATKOZAT

Név: Kancsár Enikő

ELTE Természettudományi Kar, szak: Matematika BSc

NEPTUN azonosító: A0C0M7

Szakdolgozat címe:
Humánpóz-becslés

A **szakdolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2022.05.30.

Kancsár Enikő

a hallgató aláírása

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

TERMÉSZETTUDOMÁNYI KAR

Humánpóz-becslés

SZAKDOLGOZAT

Kancsár Enikő

Matematika BSc

Matematikai elemző szakirány

Témavezető: Lukács András

Számítógéptudományi Tanszék

Budapest

2022

„All models are wrong, but some
are useful.”
George E. P. Box

Köszönetnyilvánítás

Szeretnék köszönetet mondani mindenek és mindenki előtt témavezetőmnek, Lukács Andrásnak minden útmutatásért és a rengeteg segítségért. Szeretném megköszönni Csala Barbarának a sok tanácsadást, főnökömnek, Kiss Mártonnak a rugalmasságát és pártolását, Hell Gábornak a türelmét, megértését és támogatását.

Tartalomjegyzék

1. fejezet

Bevezetés

Szakdolgozatom tárgya két cikk feldolgozása, melyek ugyanazon szerzőktől származnak és egymást követik. Ezek a 2020-as évi UniPose [3] és a 2021-es OmniPose [2] emberi pózbecslési modellek.

Az emberi pózbecslés a gépi tanulás, illetve a mélytanulás egy alkalmazási területe. A számítógépes látás (*computer vision*) feladatkörébe tartozik, ahol önálló kutatási ágként tekinthető a videópózbecslés, a tevékenységfelismerés és a jelenetmegértés alfeladatának. Célja neurális hálók tanítása az emberi ízületek és a köztük lévő kapcsolatok felismerésére, azonosítására, ezen keresztül a testhelyzetek meghatározására. Mindez számos felhasználási lehetőséget rejt magában többek között az egészségügy, ember-számítógép interakciók, videójáték-fejlesztés és a sport területén.

Az emberi pózbecslés feladata együtt nehezedik az egy képen szereplő, felismerendő személyek számával. A tanított hálónak külön kihívás helyesen összekötni, hogy az azonosított ízületek melyik személyhez tartoznak. Nevükhöz illően a UniPose modell egyetlen személy, az OmniPose akár több személy felismerésére is képes egyetlen képen. Többszemélyes pózbecslés esetében felerősödve jelenik meg egy megoldást nehezítő tényező, az *occlusion*, avagy átfedés. Ez az a jelenség, amikor egy vagy több felismerendő személy, általános esetben objektum nem látható teljes egészében a képen, fedésben, takarásban van. Okozhatja egy nem felismerendő tárgy, de takarhatják egymást is a felismerendő személyek. További nehezítő tényező a váratlan emberi pózok, például sporttevékenységek közben; a számtalan lehetséges, akár kifejezetten zsúfolt háttér; a valóságban nagy távolságra lévő emberek különböző skálával megjelenése a képen; vagy akár a lényegesen

kisebb mértékben tanulmányozott nem RGB vagy RGBD képek feldolgozása.

A többszemélyes pózbecslés alfeladatai: a kép pixeleinek klasszifikálása konkrét ízületként vagy háttérként, a képen szereplő személyek számának meghatározása és az azonosított ízületek korrekt emberi pózokba rendezése [18].

1.1. ábra. Többszemélyes emberi pózbecslés átfedésekkel, erősen hiányos személyekkel [2].

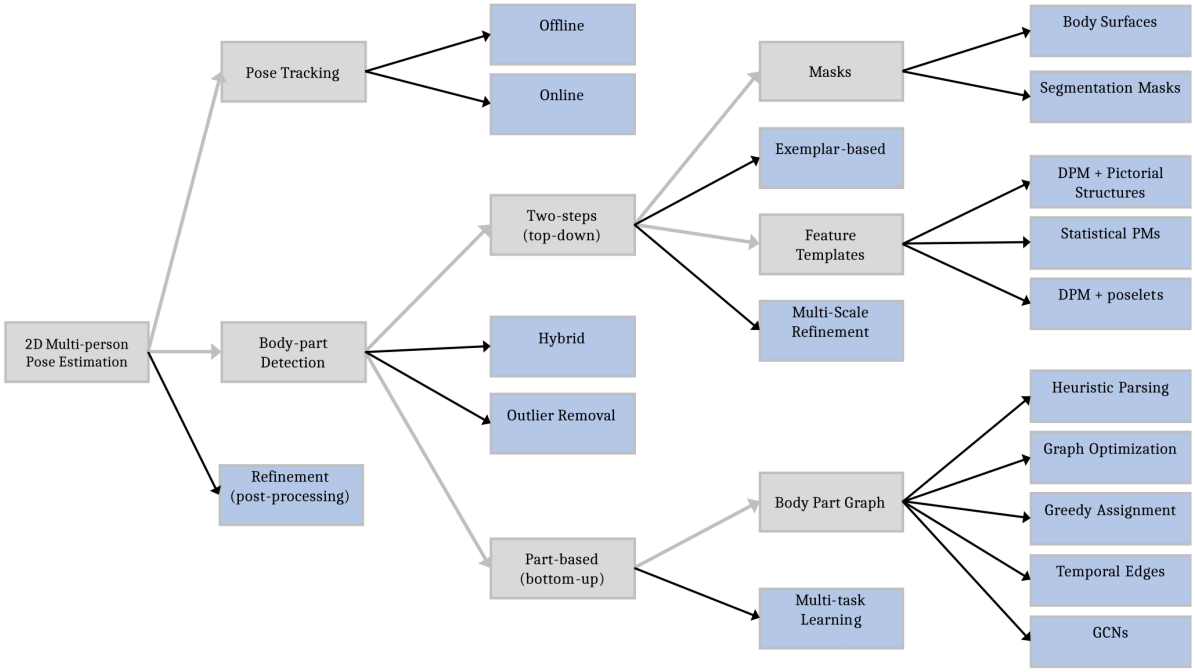
Szakdolgozatomban először bemutatom az emberi pózbecslés egy taxonómiáját, fő szempontként szem előtt tartva a tárgyalt két modell hovatartozását. Ezután az első fejezetben részletesen ismertetem a UniPose modell szerkezetét, amelyhez hozzátartozott a rendelkezésre álló nyilvános forráskód aktualizálása. A második fejezetben részletezem az OmniPose modell szerkezetét. Ez a modell megjelenésekor és a szakdolgozat megírásáig a legkorszerűbb (*state-of-the-art*, SOTA) a többszemélyes pózbecslés területén az LSP adatbázison mérve [14], amely egy elfogadott összehasonlítási alap (*benchmark*) a szakterületen.

A gépi tanulás, és még inkább a mesterséges intelligencia szakterületén a magyar szaknyelv kialakulása sajnos még gyerekcipőben jár. Bár más tekintetben természetesen pozitív, a magyar szaknyelv fejlődését hátráltatja, hogy a terület rohamosan fejlődik, rengeteg új technológiát fejlesztenek ki évről évre. Ezen kívül a köznyelvi használatban hasonló jelentésű szavakat új, eltérő szaknyelvi jelentéssel feltölteni kezdetben félreértésekhez vezethet. Ezen okoknál fogva a dolgozat nyelvhasználata némiképp váltakozni fog, első helyre helyezve az egyértelműséget, és csak ezután, de ezt szorosan követve a magyar szaknyelv felhasználását.

1.1. Taxonómia

Ebben a szekcióban a pózbecslések egy lehetséges taxonómiáját ismertetem a [18] összefoglalócikk alapján.

A pózbecslés történhet ún. markerekkel vagy azok nélkül. Ezek használata nagyrészt koncentrálódik az animációkészítésben, költségesek és tipikusan nem működnek jól pózok egy leszűkített körén kívül. A dolgozatban markereket nem használó pózbecslésről lesz szó. Három alterületet azonosíthatunk, ezek egyike a pózkövetés (*posetracking*) videókban, egy másik a *refinement*, pózfinomítás, mely adott pózok megtanulását jelenti. A dolgozatban tárgyalt modellek a harmadik területről származnak, ez az ízületfelismerés (*body part detection*). Az ízületek, testrészek (*body part, joint, keypoint*) egy előre meghatározott listából kerülnek ki, melyek adatbázisról adatbázisra változnak, bár alapvető ízületekben megegyeznek (pl. reprezentálják a fejet, a csípőt, a bokákat).



1.2. ábra. A többszemélyes emberi pózbecslés taxonómiája [18].

Két fő megközelítés különíthető el az ízületfelismerésben, a *top-down* és a *bottom-up* (fentről-le, ill. lentről-fel megközelítés). A fentről-le megközelítésű modellek rendelkeznek egy egyszemélyes pózbecslő modullal, amely azonosítja a képen szereplő embereket,

majd feltételezve, hogy minden személy azonosítva lett, megbecsüli az ízületeket. Ezáltal a fentről-le modellek tisztán kétlépcsősek, amely tulajdonságot néha elnevezésként is használják a módszerre (*two step approach*). A lentről-fel megközelítésű modellek először azonosítják az összes ízületet, majd emberként csoportosítják őket a lehetséges konfigurációk szerint (pl. egy bal könyökként azonosított ízület, nem kapcsolódhat egy jobb térdként azonosított ízülethez, csak egy bal csuklóhoz vagy egy bal vállhoz), ez a két feladat azonban nincs élesen elkülönítve a modell szerkezetében, ezért nem tekinthetőek tisztán kétlépcsősnek. Azokat a lentről-fel modelleket, amelyek egy komplett személyfelismerő modult implementálnak, a tárgyalt taxonómia hibridnek tekinti.

A fentről-le módszerek kihasználják az egyszemélyes pózbecslésben elért sikereket. Előnyük, hogy jól kezelik a különböző skálájú emberek jelenlétét (pl.: 1.1 ábra sárga váz és kék váz), mivel az egyes személyek határoló dobozaira (*bounding box*) vannak megszorítva. Hátrányuk, hogy erősen támaszkodnak a pózfelismerés elérhető pontosságára. A futásidő lineárisan nő az emberek számának növekedésével. Zsúfolt képeken, ahol az egyes személyek határoló dobozai átfedésben vannak, nem teljesítenek jól. Két népszerű megközelítés emelkedik ki, az egyik a *feature templates*, a másik a *multi-scale refinement*, többskálájú finomítás, amely egy- és többszemélyes pózbecslésben is tipikusan a legjobb eredményeket hozza a megközelítés típusai közül. Ezek a modellek a felismerendő jellemvonás (*feature*) szerint skálázást alkalmaznak, pl. egy, az egész testre egyszerre vonatkozó tulajdonság más skálát igényel, mint a kéz mozdulatai vagy az arc mimikája.

A lentről-fel módszerek között népszerű megközelítés a modell felvértezése kinematikai tudással a lehetséges pózok számának redukálására. Ezek a modellek, amelyek a páronkénti ízületkapcsolatokra helyezik a hangsúlyt, alkotják a *heuristic parsing*, heurisztikusan elemző módszerek csoportját.

Egy másik lehetséges fókusz a detektált ízületek sűrű gráfját ritkítani. Ezek a modellek *graph optimization* név alatt csoportosíthatók. Ekkor ki kell számolni egy bizonyosságot (*confidence*), amely számnak azt kell kifejeznie, hogy az adott ízületkapcsolat mennyire valószínű. Ez a mérőszám többféleképpen definiálható és az ilyen megközelítéshez megfelelő optimalizáló algoritmust és alkalmas konfidencia-mérőszámot kell választani. A mohó algoritmus valamilyen módosítását használó módszerek tekinthetők még önálló csoportnak.

A *multi-task learning* (többfunkciójú tanulás) kategorizálható lentről-fel módszerként, mivel nem alkalmaz személyfelismerő modult, viszont különbözik a többi módszertől ab-

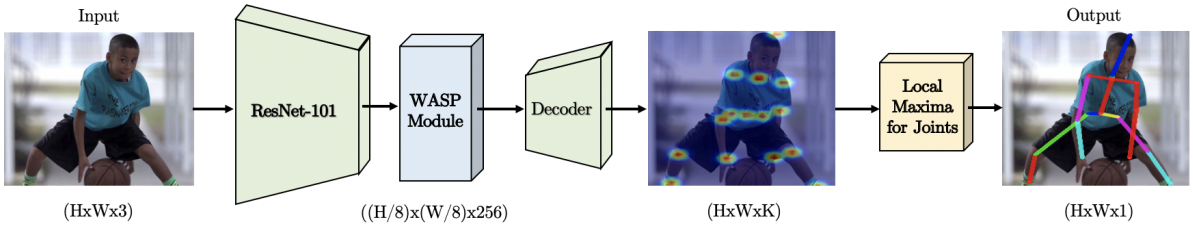
ban, hogy az ízületfelismerés és azok személyekké csoportosítása szinte egyáltalán nem különül el, erre hivatkoznak *end-to-end training*-ként.

A bemutatott taxonómia nem lehetséges, hanem ténylegesen alkalmazott módszerek alapján készült. Sok modell nem sorolható teljesen tisztán egy alkategóriába, új fejlesztések még jobban elmoshatják a határokat, vagy akár teljesen új irányzatokat fejleszthetnek ki.

A UniPose egyszemélyes pózbecslésre lett kifejlesztve, így a tanulmányozott taxonómiának szigorúan véve nem tárgya. Megengedően tekintve látni fogjuk majd, hogy a feldolgozási folyamat nem rendelkezik egyértelműen elkülönült személyazonosítási modullal, ezzel egyértelműen kizárva a fentről-le és hibrid kategorizálást. Az OmniPose *end-to-end trainable*-ként kategorizálja magát, így a lentről-fel megközelítés többfunkciójú tanulás kategóriájába sorolható.

2. fejezet

UniPose



2.1. ábra. A UniPose szerkezete [3].

A modell az alábbi *benchmark* adatbázisok formátumához illeszkedik: BBC, LSP, MPII, PennAction. Ezek közül az MPII képbázist használtam fel a forráskóddal való munkám során, így releváns összehasonlítást kapva az aktualizált modell teljesítményéről. A UniPose bármilyen felbontású bemeneti képet elfogad, ennek tesztelésére a kiválasztott adatbázis számos felbontást tartalmaz.

Ahhoz, hogy a tanító adatot előkészítse a tanításra, a modell az adatbázis képeit és annotációit átméretezi először 368x368-as felbontásra, majd 46x46-osra. Létrehoz egy hőtérképet, hogy a tanulási eredményeit alkalmasan össze tudja hasonlítani. A hőtérkép az adatértékek egy színeket használó, vizuális reprezentációja. Minél sűrűbbek az adatértékek egy pixel körül, annál melegebb szín reprezentálja az adatot [22]. A hőtérkép elkészítéséhez a Gauss-függvényt használja a modell.

A következő hiperparamétereket és metaadatokat a publikáció nem specifikálja, a nyilvános forráskódban találhatóak meg. Valószínűsíthető, hogy a modell az eredményeit az

alábbiakkal érte el. A tanításhoz a veszteségfüggvény MSE (*mean squared error*, átlagos négyzetes eltérés):

$$\text{MSE} = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i^* - y_i)^2,$$

ahol y a modell által kiszámított érték, y^* a valós érték (*ground truth*, alapigazság), n az y kimenet dimenziója. Az optimalizáló az Adam algoritmus, a *batch* méret 8, a tanítás 100 *epoch*-on keresztül fut. A tanulási ráta (*learning rate*) megtalálható a publikációban is, kezdőértéke 10^{-4}, ez minden lépésben több másik hiperparamétertől függően kerül csökkentésre.

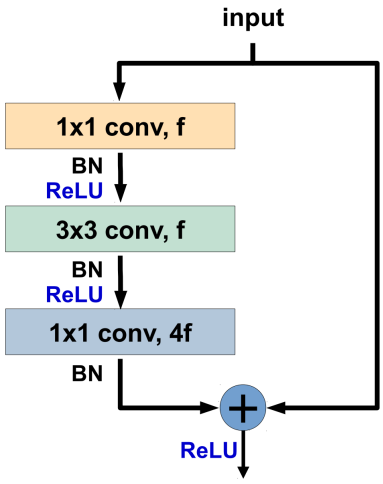
2.1. ResNet-101

A ResNet-101 a UniPose jellemvonás-felismerő (*feature extractor*) modulja. A modellnek ez a része azonosítja az egyes pixeleket a lehetséges ízületek valamelyikeként.

A *residual network* [12] megjelenése forradalmasította a gépi mélytanulást. Az intuíció azt sugallhatja, hogy egy mélyebb neurális háló több tanulásra képes és csak erőforráshiány állhatja ennek az útját. A tapasztalat ennek ellentmond, újabb rétegek szimpla egymás után halmozása nemcsak nem javít, hanem ront egy modell teljesítményén. A *residual network*, ResNet modell a nagyon mély hálók hátrányait oldja meg a *skip connection* bevezetésével. *Skip connection*-nek nevezzük azt, amikor egy réteg kimenete nemcsak a rögtön rákövetkező réteg bemeneteként szolgál, hanem egy későbbi réteg kiértékelésében is szerepe van.

Míg korábban ötven réteg mélyen teljesítményromlást lehetett tapasztalni, a ResNet modellek a *skip connection* módszert használva akár száznál is több rétegen keresztül is javítani képesek a hibaarányukat. A méretük (hiperparaméter-szám) és erőforrásigényük (flops-mennyiség) kisebb, mint például a korábbi egyik legjobb teljesítményű, hasonló célú VGG-Net hálóé [12].

A ResNet modellek *bottleneck* építőkövekből állnak. A blokk karakterizációja egy 1x1-es konvolúció után egy 3x3-as konvolúció ugyanannyi kimeneti csatornával, majd még egy 1x1-es konvolúció négyszeres kimeneti csatornával. A blokkok után *batch* normalizációt és ReLU nemlineáris aktivációt alkalmazunk kivéve a harmadik konvolúció után, ahol előbb konkatenáljuk a legelső konvolúció bemenetével.



2.2. ábra. ResNet bottleneck blokk [17, 12]. BN: batch normalizáció. f: csatornaszám.

A UniPose először végrehajt egy 7x7-es konvolúciót, melynek kimenete 64 csatorna. Ennek a konvolúciónak a lépésköze (*stride*) 2, a margója (*padding*) 3. A lépésköz azt jelenti, hogy egy konvolúció kernele hány pixelt ugrik két lépés között. A hagyományos konvolúció lépésköze 1. A margó a bemeneti kép széleihez hozzáadott pixelek száma. A hagyományos konvolúció margója 0. Egy harmadik hiperparaméter a dilatáció (*dilation*), amelyet a következő modul leírásakor részletesen ismertetek. A hagyományos konvolúció dilatációja 1. Mindhárom hiperparaméter beállításának célja a kernel látóterének növelése, a kontextus megértésének javítása.

A kimeneti kép felbontása a hiperparaméterek figyelembevételével az alábbi képlet szerint számítható ki

$$O = \frac{I - (K-1) \cdot D + 2P - 1}{S} + 1,$$

ahol O a kimenet mérete, I a bemenet mérete, K a kernel mérete, S a lépésköz, P a margó, D a dilatáció.

A kezdeti 7x7-es konvolúció után *batch* normalizáció, majd ReLU aktiváció következik, majd egy 3x3-as, 2-es lépésközű maximum *pooling* réteg. Ezután 4 csoportban ismétlődnek *bottleneck* blokkok.

Az első csoport 3 db, 64 bemeneti csatornájú blokkból áll. Ennek a csoportnak a végső kimenete nemcsak a második csoport bemeneteként szolgál, hanem *skip connection* módon átugorja a hátralévő 3 csoportot és még a UniPose következő modulját is, extra bemenetként szolgálva a harmadik, utolsó modulnak, a Decodernek.

A második csoport 4, a harmadik 23, a negyedik 3 *bottleneck* blokkból áll. A négy csoport összesen 33 *bottleneck* blokkot tartalmaz, amik egyenként 3 konvolúciót tartalmaznak, ami 99 réteget jelent. Az első csoport előtt végrehajtódik egy konvolúció és egy *max pooling*, így adódik össze a ResNet-101 101 rétege. A forráskód alapján felfedezhető, hogy a modell építése során a modult kipróbálták egy ResNet-50 és egy ResNet-10 modellel is, amelyek nem kerültek bele a végleges publikációba.

Az egyes csoportok *bottleneck* blokkjaiban a középső, 3x3-as konvolúciók rendelkezhetnek a korábban ismertetett hiperparaméterekkel is. Ezeket a publikácó nem részletezi, a nyilvános forráskód legutolsó frissítésekor az alábbiak az alapértelmezett értékek.

csoport	lépésköz	dilatáció	margó
1.	1	1	1
2.	2	1	1
3.	2	1	1
4.	1	2	2

A legutolsó 1x1-es ResNet konvolúció 2048 csatornás kimenetet küld tovább a következő modulnak, a WASP-nak.

2.2. WASP

A WASP modul a cikkírók egy korábbi fejlesztése [4]. A betűsző a *Waterfall Atrous Spatial Pooling* név rövidítése. Feladata a szemantikus szegmentáció. Ez a számítógépes látás egy önálló alterülete, nemcsak részfeladata az emberi pózbecslésnek. A bemeneti kép minden pixelének felcímkézését jelenti, ezzel olyan *szegmensekre* osztva a képet, amelyek *szemantikusan* összetartoznak.

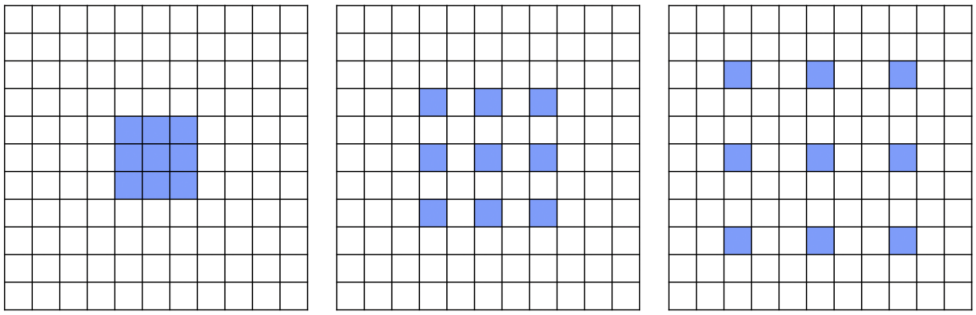
2.3. ábra. Példa szemantikus szegmentációra [4].

A pózbecslési modellek munkafolyamatának gyakori része a dekonvolúció, a bemeneti kép visszanagyítása. A tanulás során a neurális hálók egyre kisebb felbontásúra méretezik újra a képeket a predikciójuk kiszámolásáért. Feladat az eredeti bemenet felbontásához minél közelebbi kimeneti felbontásban biztosítani a predikciót.

Az egyszerű dekonvolúciós technika helyett fejlesztették ki az ASPP (*Atrous Spatial Pyramid Pooling*) modellt [6], amelynek új továbbfejlesztése a WASP [4]. Az angolosított *atrous* a francia *algorithme à trous* kifejezésből származik, ahol a francia *trou* szó lyukat jelent. Egy másik elterjedt megnevezése a *dilated*, dilatált, azaz kitágult, kitágított konvolúció. Legyen i a pixel, $x[i]$ a dilatált konvolúció bemenete a pixelre, w a szűrő/kernel, K ennek a hossza. Ekkor a konvolúció $y[i]$ kimenete:

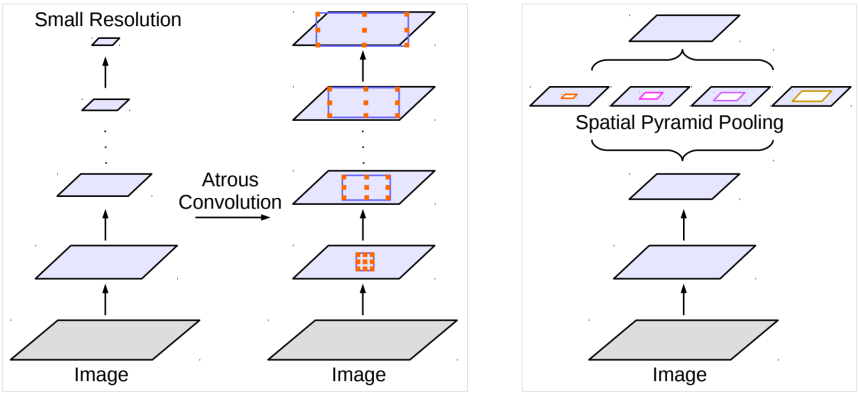
$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k]w[k],$$

ahol r a dilatált konvolúció rátája [4, 6]. A hagyományos konvolúció így a dilatált konvolúció egyrátájú speciális esete.



2.4. ábra. Dilatált konvolúció, ráta: 1, 2, 3 [4].

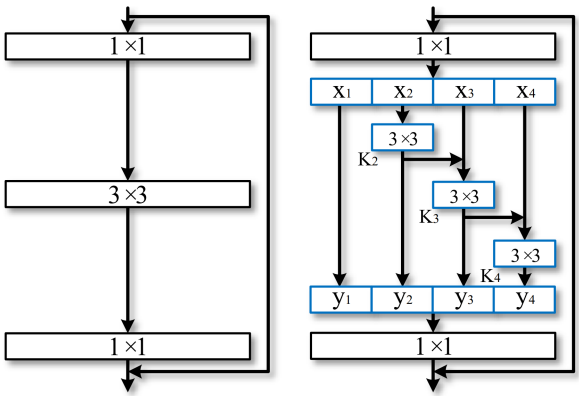
A technika bevett gyakorlat a jelfeldolgozás terén [6], ahonnan átvették és elkezdték alkalmazni a szemantikus szegmentációs feladatokra. A felfújt kernel megnövekedett látótérrel (*FOV, field of view*) rendelkezik, ezáltal növeli, avagy jobban megőrzi a kimeneti felbontást. A dilatált konvolúció felhasználásával lett kifejlesztve a *spatial pyramid pooling*, SPP technika (*térbeli, piramis típusú gyűjtőréteg*), amelynek ötlete, hogy különböző konvolúciókat egyszerre hajt végre és konkatenálja az eredményüket. Az ASPP modell az SPP technikát kombinálja különböző rátájú dilatált konvolúciókkal.



2.5. ábra. A dilatált konvolúció és az SPP technika segít megőrizni a magasabb kimeneti felbontást [7].

A zuhatagtechnika (*cascade*) párhuzamosítás és konkatenáció nélkül, egymás után hajt végre egyre növekvő rátájú dilatált konvolúciókat.

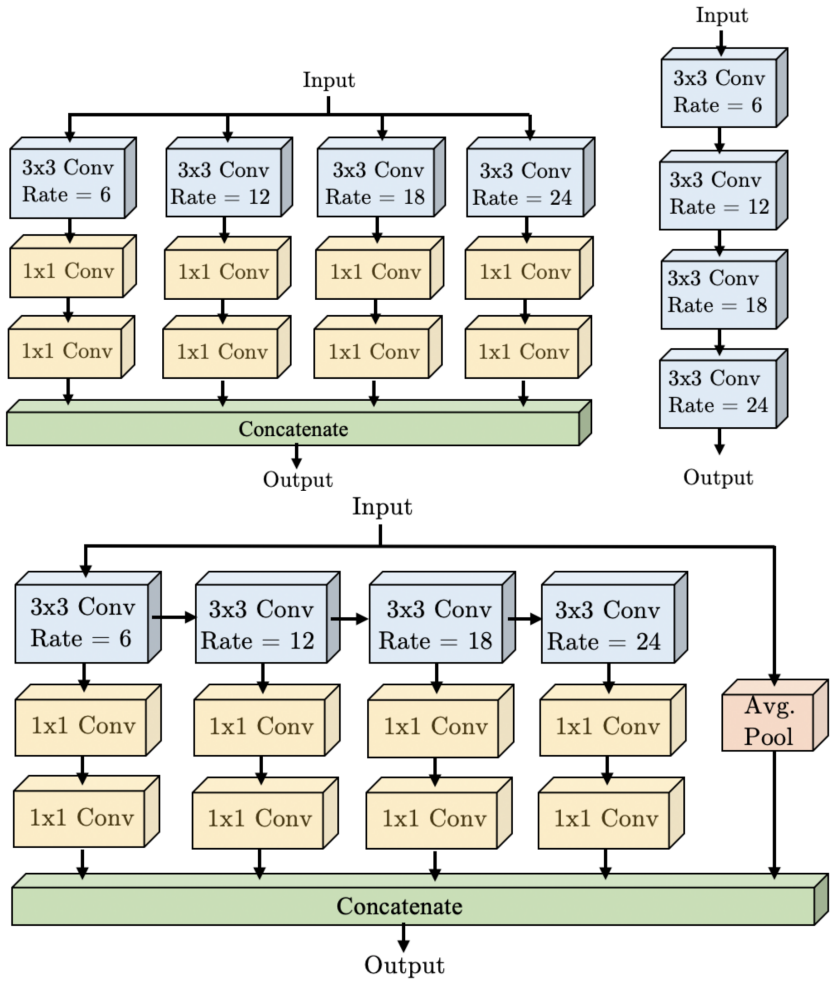
A WASP ezt a két technikát, az SPP-t és a zuhatagot kombinálja egy új, vízesésként elnevezett módszerré (*waterfall*). Ehhez az összekombináláshoz a Res2Net [11] építőkockáját használja fel, mely a ResNet már tárgyalt *bottleneck* blokkját fejlesztette tovább. Először a középső konvolúció bemenetét több csoportra osztja. A csoportok száma a Res2Net blokk skálája, skáladimenziója. Az egyik csoportra alkalmazza a ResNet blokk középső konvolúcióját, amelynek eredménye bemenetként szolgál a következő ugyanilyen konvolúciónak közösen egy következő csoporttal. Ez ismétlődik, amíg minden bemenetcsoport fel nem lett dolgozva. Végezetül minden eddigi, csoportra vonatkozó konvolúció kimenetét konkatenálja és alkalmazza rá a ResNet blokk végéről az 1x1-es konvolúciót.



2.6. ábra. Balra: ResNet Bottleneck blokk. Jobbra: Res2Net blokk, skáladimenzió: 4 [11].

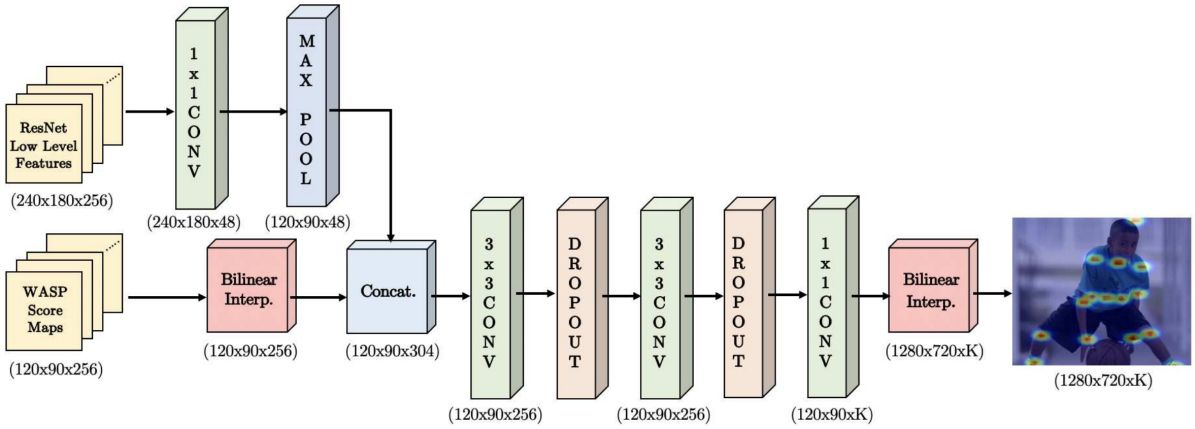
A WASP egyszerre fejleszti tovább az egymástól független Res2Net és ASPP / Cascade modelleket. A zuhatagmódszer egymás utáni, egyre növekvő rátájú dilatált konvolúcióinak kimenetére külön, egyenként alkalmaz két, egymás utáni egydimenziós konvolúciót, amiket az SPP módszer szerint konkatenál egymással, illetve még egy ún. *average pooling* réteggel, amely átlagolja a szűrőjén áthaladó értékeket.

A cikkben jelentett rátasorrend 6, 12, 18, 24. A nyilvános forráskód tanulmányozása során felfedezhető növekvő helyett csökkenő ([24, 18, 12, 6], ill. [48, 36, 24, 12]) és egyenletes ([6, 6, 6, 6]) sorrend. Ezek a lehetőségek egyértelműen felmerültek a modell tanítása során, mégsem végleges részei egyik érintett modellnek sem [3, 4, 7, 6], ami arra enged következtetni, hogy gyengébb eredményt biztosítanak, mint a növekvő rátasorrend.



2.7. ábra. Az ismertetett szerkezetek vizualizációja. Fent balra: SPP, fent jobbra: zuhatag, lent: vízesés [4].

2.3. Decoder



2.8. ábra. A Decoder modul felépítése 1280x720-as felbontású bemeneti képpel [3].

A Decoder bemenete két forrásból áll. 256 csatorna a ResNet-101 1. csoportjának kimenete, ezeket egy 1x1-es konvolúció, batch normalizáció, ReLU aktiváció és egy, a felbontást igazító *max pooling* réteg dolgozza fel először. A bemenet másik forrása a WASP modul kimenete, szintén 256 csatornával. Ez a bemenet egy bilineáris interpoláción esik át. A bilineáris típusú interpoláció a keresett pixel négy legközelebbi szomszédját használja fel egy távolság szerint súlyozott átlagszámítás közben

$$f(x, y) = a_{00} + a_{10}x + a_{01}y + a_{11}xy.$$

A fenti feldolgozások után a két bemenet konkatenálódik és a 2.8 ábrának megfelelően alkamazódnak rá az utolsó konvolúciók. Egy 3x3-as hagyományos konvolúció, batch normalizáció, ReLU, egy *dropout* réteg, majd ugyanez megismétlődik, végül egy 1x1-es konvolúció és egy 2 lépésközű *max pooling* réteg adja vissza a hőértékeket a kimeneti hőtérképhez. Hogy az ízületek hőtérképéből az emberi váz vizualizációjának pontjait megkapjuk, egy lokális maximum operáció végrehajtása szükséges.

A kimenet K darab hőtérkép értékei, ahol K az adathalmaz által meghatározott ízületek száma. A K értéke nem univerzális az adatbázisok között, a UniPose által felhasználtakban 7 és 16 között változik. Az általam felhasznált MPII képbázis 16 ízületet azonosít: két boka, két térd, két csípő, két csukló, két könyök, két váll, medence, mellkas, nyak teteje, fej.

A K darab ízület szerinti hőtérképek száma nincs összhangban a publikált forráskóddal, amely produkál még egy hőtérképet, melyet a pontossági mérőszámok átlagának kiszámításához használ fel. A cikken és a forráskódon kívül a GitHubon publikáltak a modell által megtanult súlyok, melyeket előretanításhoz lehet felhasználni. Ezek a súlyok nem működnek a forráskód $K + 1$ kimeneti hőtérképével, létrehozásukhoz a publikáció K darab hőtérképe volt szükséges.

2.4. Eredmények

A modell egyik erőssége, hogy nem használ előre definiált pózokat (*anchor poses*), amik limitálnák az általánosítási képességeit és az előre nem látott pózok megtanulását. A modell egy további képessége a *bounding boxok* (határoló dobozok) kiszámítása. A *bounding box* egy olyan téglalap egy képen, amely a lehető legszűkebben tartalmaz egy objektumot. Ezek felismerése egy külön feladat az emberi pózbecsléstől, a modell viszont a tanulási folyamata során meg tudja határozni őket, külön erre a részfeladatra szakosodott ág nélkül.

A modellnek elkészült egy csak kis mértékben módosított változata is, a UniPose-LSTM, mely videópózbecslésben 99,3%-os state-of-the-art eredményt ért el a PennAction adatbázison [21]. A módosított szerkezet a videó egy képkockájának (*frame*) pózbecsléséhez a Decoder hőtérképén kívül felhasználja az előző képkocka hőtérképét is. Miután az LSTM modul ezt feldolgozta, a kimenet még átesik öt konvolúción, hogy meghatározza a végső hőtérképet. Az utolsó két konvolúció 1x1-es, az első három esetében azonban a publikáció és a forráskód ismét eltér, előbbi 3x3-as konvolúciókat közöl, aminek a forráskódban nincsen nyoma, ehelyett 11x11-es, ötös margójú konvolúciókat használ.

Egy modell teljesítményét többféle mérőszámmal lehet mérni. Ennek kiválasztása az alapján történik, hogy a választott *benchmark* adatbázisokhoz milyen mérőszámot használnak a teljesítmény nyomon követésére. Az általam is használt MPII adatbázison a PCKh mérőszámot használják. A PCK a *Percentage of Correct Keypoints* rövidítése, amely akkor tekint egy ízületfelismerést helyesnek, ha a tanító adatban megjelölt választól vett távolsága egy bizonyos küszöbszámon belül van. Az MPII-hoz tartozó eredmények összevetéséhez egy gyakran használt küszöböt, a PCKh@0.5 jelölésűt használták, mely a fej átmérőjének 50%-át használja küszöbként. Ezzel a küszöbbel a UniPose 92,7%-ot ért el az egyszemélyes pózbecslés feladatában. Az MPII képbázis képei sok esetben nemcsak egy személyt tartalmaznak, ekkor a hivatalos annotációkban középsőként megjelölt személy

adataival vetették össze az eredményeket. A PCK mérőszámcsoport egy másik gyakori mérőszáma a PCK@0.2, mely a torzó átmérőjének 20%-át használja küszöbszámként. Ezt az LSP adatbázis használja, amelyen a UniPose elérte a state-of-the-artot, amely 94,5% volt. Ezt azóta túlszárnyalta az OmniPose.

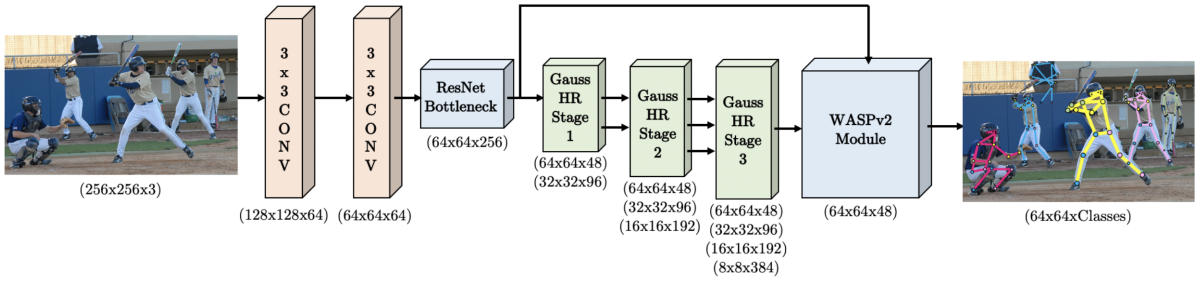
Az LSP adatbázison használatos még egy másik mérőszám is, a PCP, *Percentage of Correct Parts*, amely egy végtagot (*limb*) azonosítottnak tekint, ha a hozzá tartozó két ízület távolsága egy küszöb alatt van. A küszöbszámot a UniPose méréseihez a végtag felének választották, PCP@0.5-ként jelölve ezt. Az LSP adatbázis ízületei a csukló, könyök, váll, boka, térd, csípő, mindegyikből jobb és bal, továbbá nyak és fejtető. A PCP torzít (*bias*), mivel a természetes módon rövidebb végtagok (pl. a torzó rövidebb mint egy kar) felismerésének értékeléséhez értelemszerűen kisebb küszöbszám fog tartozni, tehát ezeknél a hibát jobban bünteti. A UniPose leírásában [3] nem egyértelmű, hogy a csípőket és vállakat csak a végtaghoz, csak a torzóhoz, vagy mindkettőhöz tartozónak tekintik-e a mérőszám számításakor. A forráskódból [5] sem deríthető ki, ugyanis abban a cikkben leírtakkal ellentétben nincs nyoma a PCP kiszámításának, helyette egy másik, az AP mérőszám található, amelyet a következő fejezetben mutatok be.

A UniPose implementálásakor és a mérésekhez PyTorch 1.0-t használtak Ubuntu 16.04 operációs rendszeren. A CPU Intel i5-2650, 2,20GHz, 16 GB RAM; a GPU NVIDIA Tesla V100 volt.

3. fejezet

OmniPose

Az OmniPose [2] modell 2021-ben jelent meg. A legkorszerűbb eredményeket érte el az LSP adatbázison a többszemélyes pózbecslés feladatában [14].



3.1. ábra. Az OmniPose szerkezete [2].

A modell hiperparamétereiről tudható, hogy a tanulási ráta kezdőértéke 10^{-3}, ez egy nagyságrenddel nagyobb, mint a UniPose modell kezdőrátája, és megegyezik a majd részletesen tanulmányozásra kerülő HRNet modellnek a COCO adatbázishoz használt tanulási rátájával. A UniPose-zal éles ellentétben, a HRNethez viszont hasonlóan ez a ráta kétszer kerül csökkentésre, 10^{-4}-re a 170. és 10^{-5}-re a 200. *epoch*-ban. A HRNetről tudható, hogy összesen 210 *epoch*-on keresztül fut [19]. A HRNet Adam optimalizáló algoritmust használ, hasonlóan a UniPose-hoz. Az OmniPose ezt az adatot nem említi, némi rizikóvállalás terhével feltételezhetjük az Adam használatát.

A modell bemutatása során az OmniPose több ponton kiemeli a költséghatékony konvolúciótípusok használatát.

A hagyományos konvolúció a csatornánkénti (*channelwise*) és a térbeli (*spatialwise*, szélesség-magasság dimenziókban) konvolúciót egy lépésben hajtja végre. Ekkor egy kernel egyszerre szűri a csatornák közötti és a térbeli korrelációkat. A két feladat azonban szétválasztható és a szétválasztás csökkenti a paraméterek számát és a számításigényt.

A mélység szerinti, mélységi (*depthwise*) konvolúció minden csatornára külön szűrőt alkalmaz. Erre és az Inception modell [20] szerkezeti ötleteire építve vezette be az Xception modell a mélységben szétválasztható, szétválasztott (*depthwise separable*) konvolúciót, mely egy mélység szerinti konvolúció után alkalmaz egy pontonkénti konvolúciót [8].

A szétválasztás egy másik módja a térben szétválasztott, avagy aszimmetrikus (*spatially separable, asymmetric*) konvolúció. A nagyobb kernelek, például egy 5x5-ös vagy 7x7-es, használata a nagyobb látótér miatt egy modell teljesítményére, pontosságára elméletileg kedvező lehet. Azonban ezek számításigénye aránytalanul nagy. Az Inception modellben kísérleteznek 3x3-as és 2x2-es kernelek sorozatával 1x1-es konvolúciókkal együtt használva a nagyobb kernelek költséghatékonyabb helyettesítésére [20]. Ezen felül alkalmaznak még aszimmetrikus konvolúciót, mely még nagyobb számításigény-csökkenést eredményez. Az aszimmetrikus konvolúció $n \times 1$-es és $1 \times n$-es kernelek váltakozó használatát jelenti. Ez a fajta konvolúció a gyakorlatban a korai rétegeken nem segíti jól a tanulást, azonban közepes méretűnek nevezett, négyzet alapú méretdimenziókon, amely alatt 12 és 20 közöttieket jelölnek, a 7x1-es és 1x7-es kernelek használata nagyon jó eredményeket hozott [20].

Mindkét típus ismert röviden szétválasztható (*separable*) konvolúcióként is. Az OmniPose mélység szerint szétválasztott konvolúciókat implementál számos hagyományos konvolúció helyett a számításigény csökkentésére. Az Xception által bemutatott szétválasztott konvolúciót [8] az OmniPose refaktorálásnak vetette alá, a mélység szerinti és a pontonkénti konvolúció között alkalmaz egy ReLU aktivációt. A dolgozat hátralévő részében a szétválasztott vagy szétválasztható konvolúció az ily módon refaktorált mélység szerinti szétválasztás rövidítése lesz.

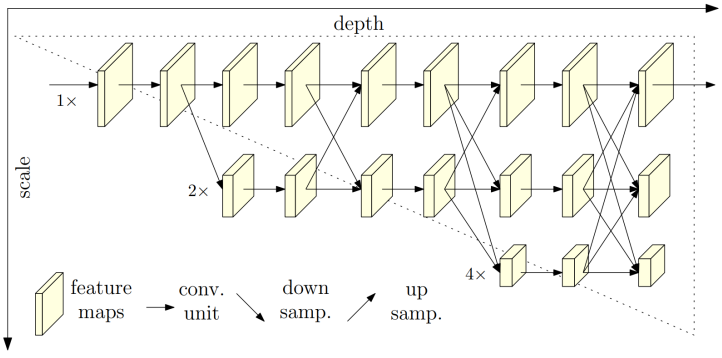
3.1. HRNet

A UniPose modell *backbone*, jellemvonás-felismerő modulja (*feature extractor*) egy ResNet-101 modell volt. Az OmniPose egy drasztikus fejlesztést hajtott végre azzal, hogy ezt a modult nem egyszerűen módosította, hanem teljes egészében lecserélte.

A HRNet, High-Resolution Net azt a célt szem előtt tartva lett kifejlesztve, hogy a teljes feldolgozási folyamat során megőrizze a bemeneti kép minél magasabb felbontását [19]. A legtöbb háló tanulási folyamata során az egyre több konvolúcióval a felbontás egyre csökken (*high-to-low resolution network*), majd az alacsony felbontású reprezentációból újragenerál egy nagyobb felbontást. A felbontásmegőrzéshez a HRNet olyan alhálókat (*subnetwork*) implementál, amelyek *high-to-low* felbontással dolgoznak. Ezeket egyesével adja hozzá a modellhez és kimenetüket többször is visszafűzi a magas felbontást végigvezető alhálójába. A magas felbontás megőrzésétől pontosabb predikciót várnak.

A HRNet a COCO, az MPII és a PoseTrack adatbázisokon lett tesztelve, melyek közül az MPII és a COCO *benchmark* adatbázisok. Megjelenésekor az MPII adatbázison megközelítette a state-of-the-art-ot [16], a COCO test-dev adatbázison pedig state-of-the-art-nak számított [10]. Megjelenése óta az OmniPose-on kívül más state-of-the-art modellek is felhasználták backbone modulként.

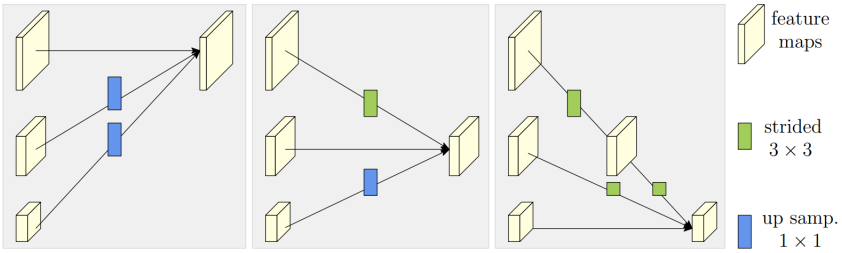
Felépítésében a HRNet egy bevett eljárást követ [19]. Először két léptetett (*strided*) konvolúció lecsökkenti a felbontást, erre a modulkezdeményre törzsként (*stem*) hivatkozunk. Ezután a fő modul elkészíti a *feature map*-eket (a kimenetet, „tulajdonságleképezések"), majd egy regresszor hőtérképeket készít az ízületekhez és visszatranszformálja a képet az eredeti felbontásra. Az OmniPose a HRNet törzsében lévő két, 2 lépésközű, de egyéb szempontból hagyományos konvolúciót lecseréli szétválasztott konvolúcióra. A HRNet új fejlesztései a fő modulra koncentrálnak, ez lesz a modellnek az a része, amely megőrzi a bemenete felbontását a feldolgozási folyamata során. Ebben a modulban más modellekkel ellentétben a *high-to-low* alhálókat nem szekvenciálisan, hanem egymással párhuzamosan helyezik el.



3.2. ábra. *High-to-high* feldolgozási folyamat [19].

A 3.2 ábrán a *conv(olutional) unit* konvolúciós egységek 3x3-as hagyományos konvolúciókat jelölnek, a *down samp(ling)* felbontást csökkentő konvolúciót jelöl, melyhez 3x3-as 2 lépésközű konvolúciót használnak. Az *up sampl(ling)* a felbontást megnövelő konvolúciót jelöl, melyet a HRNet úgy valósít meg, hogy 1x1-es konvolúcióval beállítja a csatornaszámot, majd egy legközelebbi szomszéd mintavételt használ. A HRNet tényleges felépítésében 4 párhuzamos alhálózat található, melyek előrehaladás szempontjából négy fokozatra (*stage*) bontják a teljes modellt. Az első fokozat 4 ResNet-101 típusú reziduális egységet tartalmaz, ahol az egyes egységek bottleneck blokkokból állnak, majd egy utolsó 3x3-as konvolúció zárja a fokozatot. A ResNet-50, ResNet-101 és ResNet-152 modellek építőköve mind ugyanaz, a már tárgyalt bottleneck elnevezésű blokk. A ResNet modellek közül csak a ResNet-34 blokkja különböző, amely nem nevezhető bottlenecknek [12]. A fokozatot érintő módosításokra az OmniPose leírása nem tér ki. Ebből egy lehetséges következtetés, hogy nem létezőek vagy minimálisak, másrészt a publikált ábra (3.1) alapján, amelyen megjelenik ez a modul, arra lehet következtetni, hogy egyetlen bottleneck blokkra redukálták. Ennek a blokknak a kimenete egy nagy léptékű skip connectionnel bemenetként fog szolgálni a WASPv2 modulnak. Ez a felépítés megjelent a UniPose-ban is, ahol a ResNet-101 backbone volt csoportokra osztva és az 1. csoport kimenete ugrott a Decoderhez bemenetként. Látni fogjuk majd, hogy a WASPv2-be ugrás nem jelent nagy különbséget a UniPose Decoder modulba ugrásához képest.

A HRNet további fokozatai csereblokkokból (*exchange block*) állnak, a 2., 3. és 4. fokozat rendre 1, 4 és 3 csereblokkból. Egy csereblokk 4 reziduális egységből áll, ahol ezek az egységek két 3x3-as konvolúcióból állnak az egyes alhálókban (amelyek a különböző felbontásokhoz tartoznak), továbbá egy csereegységből (*exchange unit*), mely az alhálókat köti össze. Egy csereegység minden felbontást (alhálót) mindegyikkel összeköt, úgy értve, hogy mindegyik kimenet bemenete lesz az összes alhálónak, ld. 3.2 ábra. A 3.3 ábrán láthatjuk, hogy a csereegységek hogyan aggregálják a különböző felbontásokat. Egyetlen alháló adott fokozatbeli kimenete különböző műveleten kell keresztül menjen aszerint, hogy melyik alháló bemeneteként szolgál.



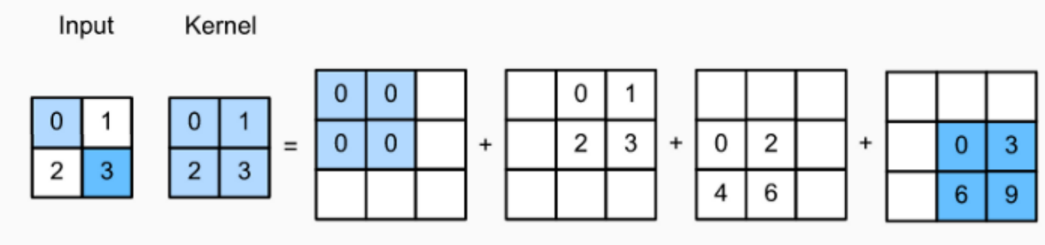
3.3. ábra. HRNet csereegység [19].

Az egy kimenetbe tartozó feature mapek tehát háromféleképpen haladhatnak tovább. Egy alháló egy adott felbontás feldolgozásáért felelős, tehát azok a feature mapek, amelyek az azonos alhálón haladnak tovább, nem mennek át extra műveleten. Amelyek egy alacsonyabb felbontású alhálóra kerülnek, azok a 3.3 ábrának megfelelően 3x3-as léptetett konvolúción mennek keresztül. A lépésköz mindig 2, nagyobb skálájú felbontáscsökkentéshez ismétlődik a léptetett konvolúció. Végezetül, azok a feature mapek, amelyek magasabb felbontású alhálók bemenetéül szolgálnak, egy 1x1-es konvolúció után legközelebbi szomszéd upsamplingen esnek át. A csereegységek biztosítják a HRNet számára a felbontásmegőrzést. Összesen 8 csereegység van a modellben, ezek 8 ún. több skálájú fúziót végeznek el (*multi-scale fusion*).

A HRNet egy kisebb és egy nagyobb verzióban lett publikálva, ezek HRNet-W32 és HRNet-W48, ahol a számok a legmagasabb felbontású alháló csatornaszámát jelölik. A W a *width*, szélesség szóból származik, mely itt a széles körben elterjedt használat helyett (ami az egyik *spatial* dimenzió lenne) a csatornaszámra utal. A HRNet-W32-ben az alhálók csatornaszáma rendre 32, 64, 138, 256. A HRNet-W48 csatornaszámai 48, 96, 192, 384. Az OmniPose HRNet-W48-at használ.

Az OmniPose-ban a HRNet utolsó három fokozatában megjelenő upsampling operációkat vetették alá módosításoknak, melyek eredményeként Gauss-HR-Stage-ként hivatkoznak rájuk (ld. 3.1 ábra). Először is transzponált (*transposed*) konvolúciót használnak. A bemenet egy pixelértékével mint skalárral megszorozza a kernelt, ezekkel az értékekkel kitölti a kimenetet a konvolúció aktuális lépésének megfelelően, az egy pixelre eső értékeket aggregálva. A kimenet méretét a lépésköz (*stride*) és a kernel méretének összehangolásával lehet beállítani, ahol ez a fajta lépésköz a kimeneten léptet. Innen ered egy másik elnevezése, a hátrafelé léptetett konvolúció (*backward strided convolution*). A transzponált konvolúció megtalálható dekonvolúció néven is, azonban ez a megnevezés félrevezető,

mivel a dekonvolúció kifejezés használatos bármilyen eljárásra, amely egy konvolúció eredményét visszafordítja vagy eltávolítja [15].



3.4. ábra. Transzponált konvolúció [15]

A transzponált konvolúció művelete után *batch* normalizáció és ReLU aktiváció következik, majd ezek után alkalmazzák még a Gauss-hőtérkép-modulációnak elnevezett műveletet egyetlen *upsampling* helyettesítésére. Ennek a módosításnak a motivációja a [23] publikáció, mely kimutatta, hogy a modellek által generált hőtérképek végső koordinátákká dekódolása az eredeti kép dimenziójába szignifikáns hatással bír egy modell végső teljesítményére. Ezért kifejlesztenek egy saját módszert, mely eloszlástudatosan reprezentálja az ízületeket (DARK, *Distribution-Aware coordinate Representation of Keypoint*). Ez a modell az OmniPose-zal egybeesően HRNetet használ backbone-ként.

A Gauss-interpoláció használatával az OmniPose jobb eredményt tud elérni, mivel az egyes kimenetek követni fogják az így elvárt Gauss-mintázatot. Egy simítást eredményez, amellyel a hamis pozitívan detektált ízületeket lehet kiszűrni. Ha f_D a bemeneti feature mapek, K Gauss-kernel, $\circledast$ egy transzponált konvolúció, akkor a kimeneti f_G feature mapek:

$$f_G = K \circledast f_D.$$

A teljes Gauss-hőtérkép-moduláció ezt a kimenetet még skálázza (f_{G_s}) az alábbi képlet szerint:

$$f_{G_s} = \frac{f_G - \min(f_G)}{\max(f_G) - \min(f_G)} * \max(f_D).$$

3.2. WASPv2

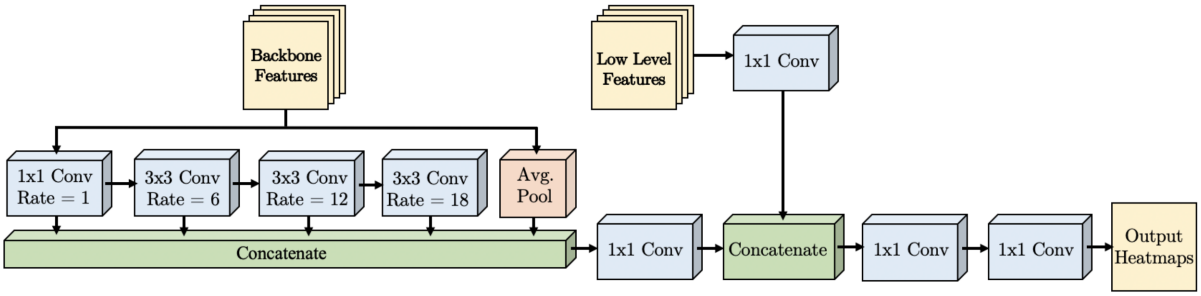
A WASPv2 modul a UniPose-ban felhasznált WASP modul továbbfejlesztése. Egy fundamentális fejlesztés a már tárgyalt szétválasztott konvolúciók és a Gauss-moduláció imp-

lementálása ebbe a modulba is, minden egyes konvolúcióba. Ezen kívül a feladatai is kibővültek, a UniPose moduljaihoz képest összevonásra került a WASP és a Decoder. Ezzel a WASPv2 feladatköre egyrészt kibővíteni a feature-kinyerést (*feature extraction*), másrészt növelni a háló látóterét (FOV) a magasabb felbontás eléréséért, megtartásáért, a pontosság növeléséért, illetve harmadrészt legenerálni a végső ízület-hőtérképeket. A hőtérkép-generáláshoz a UniPose Decoder moduljától eltérően nincs szükség interpolációra, sem *pooling* műveletekre.

A WASP-hoz hasonlóan egyre növekvő rátájú dilatált konvolúciók kimenetének konkatenációját alkalmazza a magas felbontás megőrzéséért. Szintén a WASP-hoz hasonlóan a *spatial pyramid pooling* és a zuhatagtechnika kombinációjával létrehozott vízesésszerkezetet használja. Megjelennek azonban különbségek, mint például az egyik, a legnagyobb, 24-es rátájú dilatált konvolúció elhagyása. A WASP az ASPP-ből átvetten az alábbi dilatációkkal dolgozott: 6, 12, 18, 24. Ezeket a WASPv2-ben lecserélték 1, 6, 12, 18-ra. Az *average pooling* bevételét a konkatenációba megőrizték. A különböző látótérméretet eredményező ráták használatára hivatkoznak röviden több skálájú reprezentációként (*multiscale representation*). A vízeséstechnika alkalmazása formálisan a következőképpen fogalmazható meg:

$$f_{\text{waterfall}} = K_1 \circledast \left(\sum_{i=1}^{4} (K_{d_i} \circledast f_{i-1}) + AP(f_0) \right),$$

ahol $\circledast$ a konvolúció, f_0 a bemeneti feature map, f_i az i-edik dilatált konvolúció kimeneti feature mapje, AP az average pooling operáció, K_1 1x1-es konvolúció, K_{d_i} 3x3-as konvolúció az alábbi dilatációkkal: $d_i = [1, 6, 12, 18]$.



3.5. ábra. A WASPv2 vizualizációja [2]

A UniPose Decoder moduljából örökölten egy különösen nagyot ugró *skip connection* révén a backbone által még épp csak feldolgozott feature mapek is bemenetként szolgálnak

a WASPv2 számára, melyeket némi feldolgozás után szintén a UniPose Decoder modulhoz hasonlóan konkanetál a WASPv2 kimenetével. Ehhez a konkatenációhoz a vízesésszerkezet kimenete is átesik még egy apró feldolgozáson, egy 1x1-es konvolúción. A képlet a vízesésszerkezet kimenetét ezzel az 1x1-es konvolúcióval együtt adja meg, K_1-gyel jelölve azt.

Továbblépve hasonlítsuk össze részletesebben a UniPose Decoder modult a WASPv2-ben összevont szerkezettel. A kétféle bemenetének konkatenációjához a WASPv2 egy-egy 1x1-es konvolúciót hajt csak végre a bemeneteken. Ez azt jelenti, hogy a Decoder modulhoz képest elhagytak egy max poolingot az alacsonyabb feldolgozottságú bemenetekről, illetve kicserélték a vízesés bilineáris interpolációját. A végső hőtérképek generálásához ezután szintén csak 1x1-es konvolúciókat alkalmaznak, két darabot, amelyek közül a második, utolsó képezi le a feature mapek számák az aktuális képbázis ízületszámára. Ez egy egészen drasztikus egyszerűsítés a Decoder modul több konvolúciójához, dropout rétegéhez és még egy bilineáris interpolációjához képest. Ezt a HRNet használata teszi lehetővé, amely megőrzi a magasabb felbontást a feldolgozás során, így az OmniPose-nak nincs szüksége egy teljes Decoder modulra, hogy visszanyerje a végső kimeneti hőtérképhez az eredeti képfelbontást. A teljes WASPv2 modul operációi formálisan:

$$f_{\text{WASPv2}} = K_1 \circledast (K_1 \circledast (K_1 \circledast f_{\text{LLF}} + f_{\text{waterfall}}),$$

ahol f_{LLF} az alacsonyabb feldolgozottságú bemenetet jelöli (*low-level feature maps*), mely a Gauss high-resolution szakasz első bemenetével egyezik.

3.3. Eredmények

Az OmniPose számos új kutatási eredményt egyesített egyetlen modellben. Ezzel rendkívül jó eredményeket produkált az átfedések problémájában és kiemelkedő pontosságokat ért el. Egyszemélyes pózbecslésben az LSP adatbázison 5%-ot javított a UniPose által is elért state-of-the-art eredményen [14], ezzel 99,5%-ot elérve a PCK@0.2 mérőszámmal. A PennAction adatbázis rövid videóin is javított a UniPose 99,3%-os state-of-the-art eredményén, 99,4%-ra [21].

Az előző fejezet végén megemlített AP mérőszám egy másik mérőszámon, az OKS-en alapul. Az OKS (*Object Keypoint Similarity*) azt méri, mennyire közel van egy detektált

ízület a tényleges helyéhez és az alábbi módon számítható ki [19, 9]:

$$\text{OKS} = \frac{\sum_i \exp(-d_i^2/2s^2k_i^2)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)},$$

ahol d_i az euklideszi távolság egy ízület predikciója és tényleges helye között, v_i az ízület láthatósága (*visibility flag*) (0, ha nincs címkézve az alapigazságban (*ground truth*), 1, ha címkézve van, de nem látható, 2 ha címkézett és látható), s az objektum skálája (a COCO adatbázis annotációiban az *area* mező négyzetgyöke [13]), k_i ízületenkénti konstans. Az OKS értéke minden ízületre 0 és 1 között van, a tökéletes predikció egyet ad, a kifejezetten rossz predikciók 0 körüli értéket. A k_i konstansok beállításával meghatározhatóak küszöbök az OKS-hez [9]. Az AP, vagy mAP (*mean Average Precision*) mérőszám az átlaga az OKS 10 különböző küszöbének 0,50-től 0,95-ig 0,05-onként. Megjegyezni való, hogy az AP-nek létezik más definíciója is, mely esetben az AP és az mAP nem esik egybe. Az OmniPose által használt COCO adatbázis a leírt értelemben definiálja.

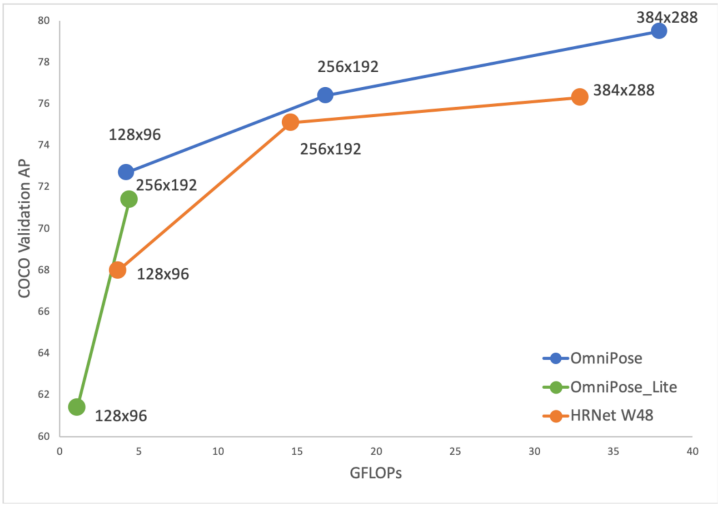
Az OmniPose nagy hangsúlyt fektet a számításigény és a paraméterszám csökkentésére. Szétválaszthatóságot építettek be a törzs léptetett és a WASPv2 dilatált konvolúcióiba. A 3.1 táblázat összefoglalja a komplexitáshoz kapcsolódó eredményeket a COCO-validation adatbázis 384x288-as bemeneti felbontású képein, a 3.2 ábra az MPII-validation adatbázison.

Modell	Paraméterszám (millió)	GFLOP mennyiség	AP
OmniPose (WASPv2)	68,1	37,9	79,5%
OmniPose (WASP)	68,2	38,6	79,2%
HRNet-W48	63,6	32,9	76,3%

3.1. táblázat. Eredmények a COCO-validation adatbázison, 384x288-es bemeneti felbontás mellett [2].

Modell	Paraméterszám (millió)	GFLOP mennyiség	PCKh@0.2
OmniPose (WASPv2)	68.1	22.6	92.3%
OmniPose (WASP)	68.2	23.0	91.2%
HRNet-W48	63.6	19.5	90.3%
OmniPose-Lite	19.4	5.8	89.0%

3.2. táblázat. Eredmények az MPII-validation adatbázison [2].



3.6. ábra. Az AP mérőszám és a GFLOP mennyiség összehasonlítása a COCO-validation adathalmazon különböző felbontások esetén [2].

A FLOP a másodpercenkénti lebegőpontos műveletek száma (*floating point operation*), a G a giga előtag rövidítése, mely egymilliárd FLOP-ot jelent. Az itt használt rövidítéskonvenció: FLOPs - Floating Point OPeration*s*; FLOPS - Floating Point OPerations *per Second*. A 3.1 táblázatban láthatjuk az ablációvizsgálat eredményeit is. Az OmniPose (WASP) modell tartalmazza a HRNetre vonatkozó fejlesztéseket, ezzel a Gauss-hőtérkép-modulációt, és nem tartalmazza a WASPv2-vel bevezetett változtatásokat. Ez egyidejűleg érvel a Gauss-hőtérkép-moduláció pontosságra gyakorolt hatása mellett az AP érték HRNethez képest vett növekedése révén, és a WASPv2 komplexitáscsökkentése mellett a csökkenő paraméterszám és GFLOP mennyiség révén.

Látható, hogy a HRNethez képest az OmniPose paraméterszáma és lebegőpontosműveletigénye öt-öttel nőtt a respektív vonatkozó nagyságrendben. A legáltalánosabban tekintve ezek a növekedések nem számítanak túl nagynak. Abszolút skálán értékelést adni nem lehet, mivel rengeteg különböző szempontot választhatunk az összehasonlításhoz: csak azokat a modelleket tekintjük, amelyek specializálódnak az átfedésre és a többszemélyes pózbecslésre, mint az OmniPose; azokat a modelleket tekintjük, melyek HRNetet használnak backbone-ként; azokat a modelleket tekintjük, amelyek egy választott pontossági mérőszám szerint egy adott adatbázison egy küszöbszámon felül vannak; azokat a modelleket tekintjük, amelyek az OmniPose-zal egy évben jelentek meg; azokat a modelleket tekintjük, amelyek az OmniPose előtt vagy után fél évvel jelentek meg; és ezek csak a

legfelszínesebb szempontok. Léteznek modellek sokkal több és sokkal kevesebb paraméterszámmal és GFLOP mennyiséggel, a két érték akár különböző irányba is változhat az OmniPose-hoz képest egyes modelleknél.

Maga az OmniPose publikációja [2] is bemutat még egy modellt, mely sokkal kisebb paraméterszámot és GFLOP mennyiséget igényel, ez az OmniPose-Lite, melyet mobilapplikációbeli integráláshoz optimalizáltak a készítők. Egy 256x256-os bemenet feldolgozásához az OmniPose 67,9 millió paramétert és 22,6 GFLOP-ot igényel. Az OmniPose-Lite ezeket 71,4 és 74,3%-kal csökkenti 19,4 millió paraméterre és 5,8 GFLOP-ra. Ezt a csökkenést úgy érték el, hogy az eredeti HRNet szerkezet minden konvolúciós rétegét szétválasztható konvolúcióra cserélték és az OmniPose WASPv2-höz hasonlóan a dilatált konvolúciókat is szétválasztottként implementálták.

A teljesítmény számításakor a UniPose-zal megegyező fejlesztői körülményeket használtak: PyTorch, Ubuntu 16.04 operációs rendszer; Intel i5-2650, 2.20GHz, 16 GB RAM CPU; NVIDIA Tesla V100 GPU.

4. fejezet

Felhasználás

A dolgozattal való munkám során részletekbe menően megismerkedtem a state-of-the-art OmniPose [2], többszemélyes pózbecslési modell szerkezetével. Jövőbeli cél a modellhez egy nyílt forráskód implementálása. Ehhez szintén részletekbe menően tanulmányoztam az OmniPose előfutárának tekinthető UniPose [3], egyszemélyes pózbecslési modellt, mely rendelkezik egy nem karbantartott, nyilvános forráskóddal [5]. Eddigi munkámnak jelentős része volt ezt a forráskódot aktualizálni a publikált szerkezethez minél nagyobb hűséggel. Ehhez felhasználtam a GitHubon feltett kérdések (GitHub Issue) közül a megválaszoltakat és a megválaszolatlanokat is, továbbá a *fork*-okat. Első lépésben ki kellett javítani pár szintaktikai hibát, kiszűrtem a kódduplikátumokat és a nem használt kódrészleteket. Ezután a kompaktabb kezelhetőségért létrehoztam egy konfigurációs fájlt, mely a hiperparamétereket és metaadatokat tartalmazza egy helyen, a kódban elszórtság helyett. Ehhez készítettem egy mintafájlt is, mely az alapértelmezett értékeket tartalmazza, ezzel rezilienssé téve a fejlesztési folyamatot a konfigurációs értékek eseti átírásából származó hibalehetőségekre. Az egyes benchmark adatbázisok különböző felépítést használnak az annotációik megadására. Ezeket valamilyen módon egyesíteni kellett a UniPose kifejlesztéséhez, azonban ez nincs dokumentálva és a forráskód alapján csak részben lehet megállapítani ezeket a transzformációkat, így a kiválasztott MPII adatbázishoz [1] kidolgoztam egy igazítást. A UniPose nem listázta a felhasznált könyvtárait és azoknak a megkövetelt verzióját (*requirements*), ami a könyvtárak egymással nem kompatibilis verzióinak telepítéséhez vezethet, ezzel ellehetetlenítve a sikeres futtatást. Megállapítottam kompatibilis verziók egy csoportját és a munkafolyamat előrehaladtával is rendszeresen dokumentáltam ezek listáját. A UniPose az eredményeket csak a terminálban jelentette

meg, szövegesen. Automatizáltam ezen adatok fájlba mentését és implementáltam ábrák készítését egy könyvtárral. A forráskódon kívül publikáltak előretanított súlyokat is, melyeket felhasználva gyorsabb áttanítás volt remélhető. Ezek a súlyok azonban nem működnek a modellel, ennek oka, hogy a forráskódban a kimeneti hőtérképek száma az adott adatbázis által felhasznált ízületek száma helyett ennél eggyel több. Aktuális feladatom ennek az eltérésnek a pontos megértése és kijavítása. Az elkészült implementáció először újabb teszteléseken fog átesni: ellenőrzés végett olyan képbázisokon, amelyekről elérhető nyilvánosan a teljesítménye, majd fejlesztésképp újabb adatbázisokon, akár egyéni gyűjtésű képhalmazokon. Meggyőződve az új forráskód teljesítményéről, elkezdhető az OmniPose szerkezet implementációja. Ebben segítséget nyújt természetesen az elkészült UniPose kód számos aspektusban, a nyilvánosan elérhető HRNet [19] forráskód, mely az OmniPose backbone-ja, és a nyilvánosan elérhető Xception [8] forráskód, mely bevezette a mélységileg szétválasztott konvolúciót. Erre a pontra eljutva több irányba is tovább lehet haladni. Kutathatók a komplexitáscsökkentés módjai, fejleszthető a teljesítmény az elismert benchmark adatbázisokon, fejleszthető a teljesítmény széles körű, teljesen új bemeneten, akár teljesen új alkalmazási területen. A képi emberi pózbecslés alapköve a videós pózkövetésnek, melynek számos hasznos alkalmazása lehetséges. Felhasználható humanoid robotok építésében, megfigyelő rendszerek kiépítésében, digitális eszközök okosfunkcióinak fejlesztésére, önvezető járművek építésében; szórakoztatásban használható videójáték-fejlesztésre, virtuális valóság kiépítésére; sportban az edzői munka segítésére játékosstratégiák analizálása által, akár saját, akár ellenfél játékosok esetében. Új diagnosztikai eszköz lehet az egészségügyben, ahol alkalmazható kora gyermekkorban fejlődéskövetésre, fejlődési rendellenességek felismerésére, felnőttkorban tartáskövetés esetén sérülések megelőzésésére, továbbá idegrendszeri sérüléssel vagy fejlődésbeli lemaradással élő páciensek állapotának kiértékelésére.

Irodalomjegyzék

[1] Mykhaylo Andriluka és tsai. „2D Human Pose Estimation: New Benchmark and State of the Art Analysis". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014. jún.

[2] Bruno Artacho és Andreas E. Savakis. „OmniPose: A Multi-Scale Framework for Multi-Person Pose Estimation". *CoRR* abs/2103.10180 (2021). arXiv: 2103.10180. URL: https://arxiv.org/abs/2103.10180.

[3] Bruno Artacho és Andreas E. Savakis. „UniPose: Unified Human Pose Estimation in Single Images and Videos". *CoRR* abs/2001.08095 (2020). arXiv: 2001.08095. URL: https://arxiv.org/abs/2001.08095.

[4] Bruno Artacho és Andreas E. Savakis. „Waterfall Atrous Spatial Pooling Architecture for Efficient Semantic Segmentation". *CoRR* abs/1912.03183 (2019). arXiv: 1912.03183. URL: http://arxiv.org/abs/1912.03183.

[5] *bmartacho/Unipose@Github.com*. https://github.com/bmartacho/UniPose. Accessed: 2022-05-04.

[6] Liang-Chieh Chen és tsai. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. 2016. DOI: 10.48550/ARXIV.1606.00915. URL: https://arxiv.org/abs/1606.00915.

[7] Liang-Chieh Chen és tsai. *Rethinking Atrous Convolution for Semantic Image Segmentation*. 2017. arXiv: 1706.05587 [cs.CV].

[8] François Chollet. *Xception: Deep Learning with Depthwise Separable Convolutions*. 2016. DOI: 10.48550/ARXIV.1610.02357. URL: https://arxiv.org/abs/1610.02357.

[9] *COCO - Common Objects in Context*. https://cocodataset.org. Accessed: 2022-05-25.

[10] *COCO test-dev Benchmark (Pose Estimation) | Papers With Code*. https://paperswithcode.com/sota/pose-estimation-on-coco-test-dev. Accessed: 2022-05-04.

[11] Shang-Hua Gao és tsai. „Res2Net: A New Multi-Scale Backbone Architecture". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.2 (2021. febr.), 652–662. old. DOI: 10.1109/tpami.2019.2938758. URL: https://doi.org/10.1109%2Ftpami.2019.2938758.

[12] Kaiming He és tsai. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[13] *Human Pose Estimation: Deep Learning Approach [2022 Guide]*. https://www.v7labs.com/blog/human-pose-estimation-guide. Accessed: 2022-05-25.

[14] *Leeds Sports Poses Benchmark (Pose Estimation) | Papers With Code*. https://paperswithcode.com/sota/pose-estimation-on-leeds-sports-poses. Accessed: 2022-05-04.

[15] Divyanshu Mishra. *Transposed Convolution Demystified*. https://towardsdatascience.com/transposed-convolution-demystified-84ca81b4baba. [Online; accessed 23-May-2022]. 2020.

[16] *MPII Human Pose Benchmark (Pose Estimation) | Papers With Code*. https://paperswithcode.com/sota/pose-estimation-on-mpii-human-pose. Accessed: 2022-05-04.

[17] Edmar R. S. de Rezende és tsai. *Exposing Computer Generated Images by Using Deep Convolutional Neural Networks*. 2017. DOI: 10.48550/ARXIV.1711.10394. URL: https://arxiv.org/abs/1711.10394.

[18] Eduardo Souza dos Reis és tsai. „Monocular multi-person pose estimation: A survey". *Pattern Recognition* 118 (2021), 108046. old. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2021.108046. URL: https://www.sciencedirect.com/science/article/pii/S0031320321002338.

[19] Ke Sun és tsai. *Deep High-Resolution Representation Learning for Human Pose Estimation*. 2019. DOI: 10.48550/ARXIV.1902.09212. URL: https://arxiv.org/abs/1902.09212.

[20] Christian Szegedy és tsai. *Rethinking the Inception Architecture for Computer Vision*. 2015. DOI: 10.48550/ARXIV.1512.00567. URL: https://arxiv.org/abs/1512.00567.

[21] *UPenn Action Benchmark (Pose Estimation) | Papers With Code*. https://paperswithcode.com/sota/pose-estimation-on-upenn-action. Accessed: 2022-05-22.

[22] Zbigniew Zdziarski. *Generating Heatmaps from Coordinates*. https://zbigatron.com/generating-heatmaps-from-coordinates/. [Online; accessed 18-April-2022]. 2022.

[23] Feng Zhang és tsai. *Distribution-Aware Coordinate Representation for Human Pose Estimation*. 2019. DOI: 10.48550/ARXIV.1910.06278. URL: https://arxiv.org/abs/1910.06278.