Óbudai Egyetem
Neumann János Informatikai Kar

3D-s arckép rekonstrukciója egy 2D-s képből mesterséges intelligencia segítségével

---

Reconstructing 3D face picture from single 2D picture supported by artificial intelligence

2022. november 26.

Témavezető:
Vámossy Zoltán
Egyetemi docens

Készítette:
Gaál Bernát Ruben - HBGCXK
Hua Nam Anh - DQ4LFK

Tartalomjegyzék

Absztrakt

A közelmúltban a mélytanuláson alapuló 3D arcrekonstrukciós módszerek ígéretes eredményeket mutattak mind minőség, mind hatékonyság tekintetében. A neurális hálózatok tanítása azonban jellemzően nagy mennyiségű adatot igényel, ami magával vonza a megfelelő erőforrásokat. A felsoroltak hiányában mi az alábbi megoldást javasoljuk. Egy gyengén felügyelt tanítású hálózatot amit, lehetséges "in-the-wild" képekkel betanítani. Ennek megvalósításához kettő már kész kutatás anyagait vettük igénybe. Az első, [32]DECA képes kezelni az arc kisebb részleteit, arckifejezéseit. A másik, a [33]FOCUS bemutatott egy korszerű megközelítést az arckép rekonstrukciójára okklúziók mellet, mint például szemüveg,sapka stb. Ezek a kutatások nem csak korszerűek, de a NOW challenge benchmark-ja alapján bizonyitottan jól teljesítenek. Ezeket ötvözve egy robosztus, illetve realisztikus arcképek generálására alkalmas módszert mutatunk be ebben a tanulmányban. A rekonstrukció mellet implementáltunk egy arckép analízálására alkalmas módszert, amely képes eldönteni az arcképen lévő személy érzelmi állapotát és korát. Ezek köré egy felhasználó barát microservice alapú web applikációt biztosítunk, melynek szolgáltatásai felhőn üzelmelnek.

1. Bevezetés

Az elmúlt évek során, egyre több figyelmet kaptak a digitális képfeldolgozáson alapuló technológiák az informatikában. Mint ahogy [5] is megvan említve az arcfelismerő technológiák széles körben elterjedtek napjainkban, beleértve a biztonságot, animációt és egészségügyet. Illetve, ezen szakmai területen mostanság felkapott, hogy 3D-s adatok implementálásával megkerüljék a 2D arckép által megszabott határokat, mivel a 2D-s kép képtelen az emberi arc geometriájának eltárolására. A 3D arcfelismerés sokkal pontosabb adatokat ad vissza, például pózban és megvilágításban, amelyek hátulütői a 2D-nek. Azonban ennek is vannak hátrányai, mint például, hogy sokkal nagyobb komplexitású képfeldolgozást igényel, ezáltal szűkítve a lehetőségeket.

Az arcokat többféleképpen is lehet rögzíteni, például Stereo-vision rend-

szerekkel, amelyek két kamerát használnak. Ezek a kamerák ugyan arról az objektumról készítenek párhuzamosan képeket, majd ezeket összehasonlítva visszaadják a képen lévő egy pontnak a mélységét.

Egy másik módszer a 3D lézerszkenner (pl. NextEngine, Cyberware), amelyet elsősorban ipari célokra fejlesztettek. Ipari termékek vizsgálatával szemben, az emberi arc feltérképezéséhez több feltételt kell figyelembe venni. Mivel az emberi arc nem lehet teljesen mozdulatlan, fontos, hogy a szkenner által készített felvétel időintervalluma csekély legyen. Röviden, a lézerszkenner fényhullámokat szór az objektumra, s ezek annak felszínéről visszaverődnek a szenzorra. A szenzor ezután kiszámítja az objektum felszínének távolságát az alapján, hogy mennyi idő alatt tette meg a teljes utat a hullám. Ezt a folyamatot "$Time\ of\ flight$"-nak szokás nevezni.

A következő technológia a 3D-s adatok rögzítésére az RGB-D kamerák (pl. Kinect) használata. Ezek olyan RGB kamerák, amelyek rendelkeznek infravörös szenzorral, mely mélységi adatokat biztosít, ezáltal egy RGB képet ad vissza, amiben minden pixelhez tartozik egy mélységi érték.

Bár, ezek a módszerek mind megfelelnek 3D-s adatok gyűjtésére, az első két megvalósítás hátránya, hogy előre megszabott feltételeknek megfelelő környezetet és drága felszereléseket igényel egy jó minőségű arc szkenneléshez. Ellenben, az RGB-D kamerák olcsóbbak és könnyebben használhatóak de a minőség korlátozott.

A fent említett megközelítések általában költséges optimalizálási folyamatot igényelnek a jó minőségű 3D-s arc visszanyerése érdekében. Két évtized telt el [8] Vetter és Blanz úttörő munkája óta, amely először mutatta be hogyan lehet egyetlen képből rekonstruálni az arc 3D-s geometriáját. Azóta a 3D arcrekonstrukciós módszerek rohamosan fejlődtek, de a korábbi modellek csak az arc durva alakját tudták rekonstruktuálni és képtelenek voltak kinyerni az arckifejezéstől függő geometriai részleteket, mint például aráncokat, amik fontosak a realisztikusság szempontjából.

Később jöttek újabb modellek, melyek képesek voltak kiragadni a fent említett geometriai részleteket, azonban hátrányuk, hogy egy nagy volumenű, jó minőségű tanító adathalmazt követelnek. Illetve, egy másik hátrányuk, hogy inkonzisztens módon teljesítettek az okklúziókkal szemben. Az okklúziók mindenütt jelen vannak, és eleve nehezen kezelhetőek az alakjuk, megjelenésük és a pozíciójuk sokfélesége miatt. Ezáltal okozott problémaaz, hogy az arcmodell alkalmazkodik az eltakart arcrégióhoz, és ennek eredményeként a rekonstruált arc torz lesz. Ezért fontos nyitott kérdés marad annak eldön-

tése, hogy mely pixelek illeszkedjenek és melyek ne illeszkedjenek az arcra egy 3D arc rekonstrukciója során okklúziók jelenlétében. A közelmúltban számos olyan módszert javasoltak, amelyek gyengén felügyelt tanítású konvolúciós neurális hálózatokat(CNN) használnak a hatékony, robosztus és a fent említett kihívásokat leküzdő arcrekonstrukció eléréséhez

Tehát, a jelenlegi monokuláris 3D arcrekonstrukciós módszerek képesek finom geometriai részleteket visszaadni, azonban számos megkötéssel küzdenek. Egyes módszerek olyan arcokat generálnak, amelyeket nem lehet valósághűen animálni, mivel nem modellezik, hogy hogyan változnak a ráncok az arckifejezésekkel. Más módszerek kiváló minőségű szkennelt arcokat használnak tanításhoz, és nem jól általánosíthatóak "in-the-wild" képekre. A Yao Feng et. al. [32] által bemutatott DECA model a legelső olyan megközelítés, amely regresszálja a 3D arcformát és az animálható részleteket, amelyek egy személyre jellemzőek, de az arckifejezéssel változnak, ezáltal képesek az arc valósághű animálására.

Az okklúziókkal szemben a legtöbb megközelítés a 3D arc illesztéséhez inverz renderelést alkalmaz egy adott eltakaró szegmentálásához. Ennek hátránya, hogy egy okklúzió szegmentációs modellhez nagy mennyiségű annotált adatra van szükség. Chunlu Li et. al [33] ezzel ellentétben egy olyan modellalapú megközelítést mutat be a 3D arc rekonstrukciójához, amely rendkívül robosztus az okklúziókkal szemben, de nem igényel semmilyen okklúziós annotációt a tanításhoz.

Ebben a tanulmányban az a célunk, hogy egy olyan animálható 3D arcrekonstrukciós modellt készítsünk gyengén felügyelt tanulással, amely robosztus az okklúzió ellen, a fent említett DECA és F OCUS modellek segítségével. Ezt kiegészítve, bemutatunk egy arcképet elemző megoldást, mely visszaadja a célszemély érzelmi állapotát, illetve életkorát. Eszrevettük a munkánk elkészítése során, hogy a meglévő megoldásokat nehéz egy átlagos személynek kipróbálnia technikai tudás hiányában, így a fenti két szolgáltatás köré egy webapplikációt készítünk, amit felhőn üzelmeltetünk.

Összefoglalva, ez a dokumentum az alábbi öt szempontot fogalmazza meg:

- Bemutatunk egy CNN-alapú, egyetlen képen alapuló arcrekonstrukciós módszert, amely kihasználja a hibrid szintű képinformációt a gyengén felügyelt tanuláshoz.
- Konzisztens működést biztosít különböző okklúzók mellet.
- Az arckifejezésektől függő geometria adatok azonosításával egy animál-

ható realisztikus 3D arcrekonstrukciót hozunk létre.

- Arc elemzésére alkalmas megoldás.
- Webapplikáció elkészítése és a szolgáltatások megfelelő mükődése a felhőn

2. Kapcsolódó kutatások

Ahogy láthatjuk, különböző munkák megkísérelték a fellépő komplex kihívásokat legyőzni, minél kreatívabb megközelítéssekkel. Ezekről tovább Vetter és Blanz [8] munkájában és további dokumentumokban lehet olvasni. Ebben a fejezetben megvizsgáljuk a Yao Feng et. al. [32] valamint Chunlu Li et. al [33] által bemutatott megközelítéseket és megnézzük, hogy miért is fontosak a mi munkánk szempontjából.

2.1. DECA

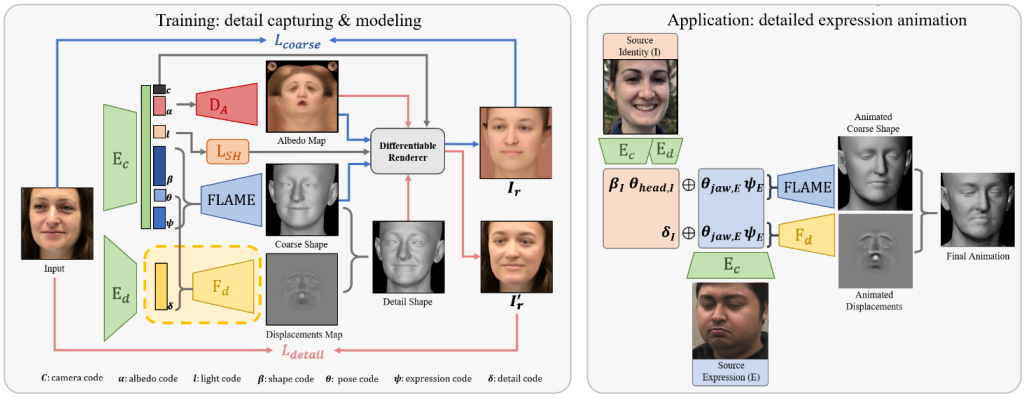
Yao Feng et. al.[32] az egyetlen, "in-the-wild" 2D-s képből rekonstruált animálható 3D arcmodell készítése érdekében fejlesztette ki a DECA (Detailed Expression Capture And Animation) modelljét.

Egy "in-the-wild" képekkel tanított animálható elmozdulási modellt javasol, amely a kifejezési paraméterek változtatásával képes hiteles geometria részletek előállítására. Az előbbi cél elérése érdekében egy újszerű részletkonzisztencia költségfüggvényt mutat be a statikus és az arckifejezésekre dinamikusan változó geometriai adatok szétválasztására.

Ammennyiben a tanítás során két kép érkezik különböző arckifejezésekkel, megfigyelhető, hogy a 3D arcformájuk és a személyspecifikus részleteik megegyeznek. Ezt a megfigyelést használják ki a részletkódok felcserélésével az azonos személyről készült különböző képek között, és kikényszerítik, hogy az újonnan renderelt eredmények úgy nézzenek ki, mint az eredeti, bemenetként átadott képek.

A geometria részletek rekonstrukciója robosztus a jellemző okklúziókra, a pózok nagyfokú változására és a megvilágítási variációkra.

A továbbiakban megvizsgáljuk a DECA modell működési elvét, architektúráját.



1. Ábra. DECA tanítás és animáció. Forrás: Forrás:[32]

2.1.1. Kódoló

Első lépésként egy durva rekonstrukciót (a FLAME modelltérben [33]) tanítanak be *analysis-by-synthesis* módon: egy bemenetként átadott I 2D-s képből látens kódot készítenek, ezt dekódolják annak érdekében, hogy egy I_r 2D-s képet hozzanak létre, valamint minimalizálják a szintetizált és a bemeneti kép közötti különbséget.

Az ábrán **??** látható módon, egy E_c kódolót tanítanak be, amely egy ResNet50 [35] hálózatból, és egy azt követő teljesen összekapcsolt rétegből áll, egy alacsony dimenziós látens kód regressziója céljából. Ez a látens kód tartalmazza a ω (identitás), ψ (kifejezés), θ (póz) FLAME [34] paramétereket(azaz reprezentálja a durva geometriát), az albedó együtthatókat α, a kamera c és a megvilágítási I paramétereket.

A F LAM E [33] egy statisztikai 3D fejmodell, amely egyesíti a különálló lináris identitás-alak és kifejezés tereket linear blend skinning-gel (LBS) ´es a pózfüggő korrektív blendshape alakzatokat, annak érdekében, hogy a nyak, az állkapocs és a szemgolyók mozgathatóak legyenek. Adott arc identitás R| |, póz R3k+3 (ahol k = 4 a nyak,´allkapocs, és szemgolyók ízületeinek száma), és kifejezés R| | paraméterek mellett, a FLAME egy n = 5023 csúcspontot tartalmazó archálót (mesh) ad kimenetül. A modell a következőképpen van definiálva: M (, ,) = W

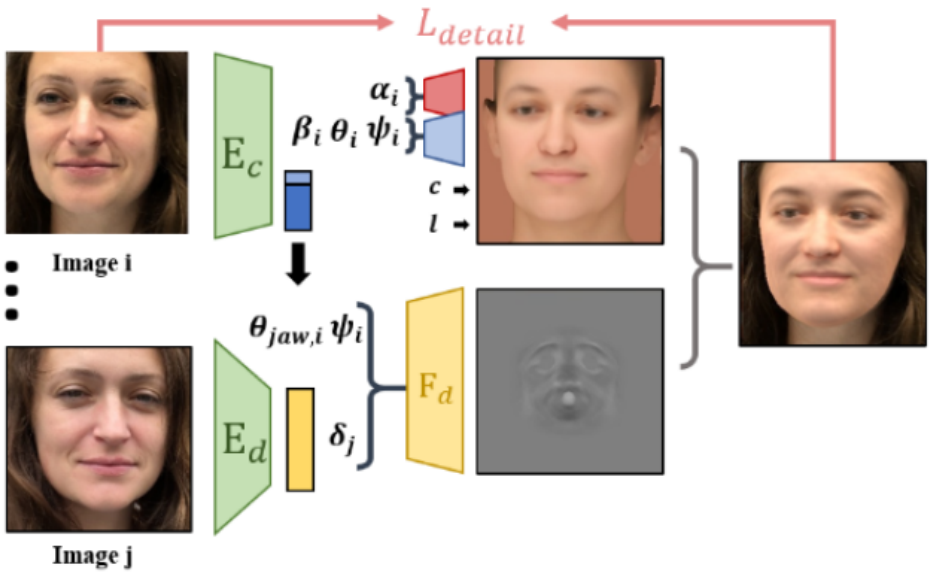
(Tp(, ,), J( ), , W) (1) , ahol a blendskinning függvény W (T, J, , W) elforgatja a T R3n csúcsait a J R3k ízületek körül, lineárisan finomítva a W Rk∗n keverési súlyokkal. A J ízületi helyek a identitás függvényeként definiálhatók. Továbbá, Tp(, ,) = T + BS (, S) + BP (, P) + BE (, E) (2) jelöli a T minta átlagát "nullpózban" a hozzáadott alak blendshape- ekkel BS (, S) : R| | R3n, pózkorrekciókkal BP (, P) : R3k+3 R3n, valamint kifejezés blendshape-ekkel BE (, E) : R| | R3n, a megtanult identitás-, póz- és kifejezésalapokkal (lineáris alterekkel) S, P és E.

2.1.2. Dekódoló

A dekódoló hálózat egy részletes UV elmozdulási térképpel D egészíti ki a durva FLAME geometriát. A durva rekonstrukcióhoz hasonlóan egy Ed kódolót tanítanak be (amelynek architektúrája megegyezik az Ec felépítésével), hogy az I 2D-s képből egy 128 dimenziós látens kódot készítsen, amely az alanyspecifikus részleteket reprezentálja. A látens kódot ezután a FLAME kifejezés és állkapocs póz jaw paramétereivel kapcsolják össze, majd Fd az imént említett paraméterekből előállítja D-t. Fd a látens kódot felhasználva szabályozza a statikus személyspecifikus részleteket. Kihasználja a durva rekonstrukcióból kapott kifejezés , és az állkapocs jaw paramétereket a dinamikus, kifejezésektől függőráncok részleteinek rögzítése érdekében. A rendereléshez D-t normáltérképpé alakítják.

2.1.3. Részletkonzisztencia veszteség

részleteinek rögzítése érdekében. A rendereléshez D-t normáltérképpé alakítják. 2.1.3 Részletkonzisztencia veszteség´Abra 2: Részletkonzisztencia veszteség. Forrás:[31] Yao Feng et. al. [31] az identitás-függő és a kifejezésektől függő részletek szétválasztása érdekében egy új részletkonzisztencia veszteség függvényt javasol. A metódus nélkül a személyspecifikus látens kód rögzíti az iden- titástól és a kifejezésektől függő részleteket. Ebből kifolyólag a rekonstruált részletek nem repozicionálhatóak a FLAME állkapocs póz jaw és kifejezés paramétereinek megváltoztatásával. Amennyiben adott két kép Ii és Ij ugyanarról az alanyról (ci = cj), a veszteséget a következőképpen határozzuk meg: Ldc = Ldetail(Ii, R(M (i, i, i), A(i), Fd(j , i, jaw,i), li, ci)) (3) , ahol i, i, jaw,i, i, és ci Ii paraméterei, valamint j Ij részletkódja.



2. Ábra. Részletkonzisztencia veszteség Forrás: Forrás:[32]

2.1.4. Tanítás

Ebben az alfejezetben a [1] ábrán szemléltetett DECA [31] pipeline-t vizsgáljuk meg. A tanítás során (bal oldali doboz) a DECA [31] minden egyes képhez az arc alakjának rekonstruálásához szükséges paramétereket becsli meg az alak konzisztenciainformáció segítségével (a kék nyilakat követve), majd a részletkonzisztenica-információ (a piros nyilakat követve) kihasználásával megtanul egy kifejezésfüggő elmozdulási modellt. A sárga doboz tartalmazza az elmozdulási konzisztencia-veszteséget, amelyet részletesebben a [2] ábra szemléltet. A tanítás után a DECA animál egy arcot ([1] ábra, jobb oldali doboz) a rekonstruált forrásidentitás alakjának, fejpózának és részletkódjának, valamint a forráskifejezés állkapocs pózának és kifejezési paramétereinek kombinálásával, annak érdekében, hogy egy animált durva alakot és egy animált elmozdulási térképet kapjon. A pipeline kimenete egy animált, részletes alakzat.

2.1.5. Adathalmazok

A DECA-t három nyilvánosan elérhető adathalmazon tanítják [31]:

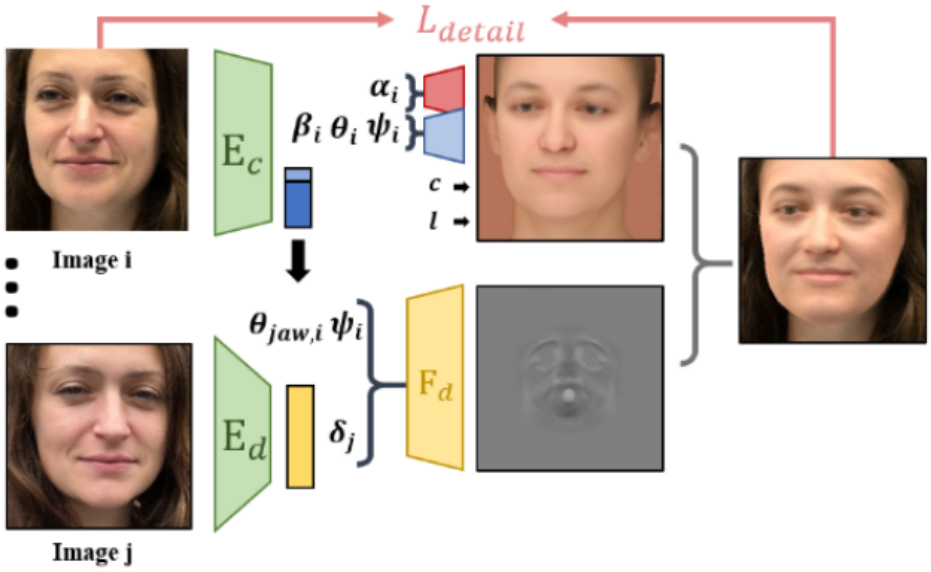
- VGGFACE2
- BUPT-Balancedface
- VoxCeleb2

A VGGFace2 több, mint 8 ezer alany képeit tartalmazza, átlagosan több, mint 350 képpel alanyonként. A BUPT-Balancedface 7 ezer képet kínál etnikumonként, és a VoxCeleb2 pedig 145 ezer videót tartalmaz 6 ezer különböző alanyról.¨Osszességében a DECA-t 2 millió képpel tanították be.

2.2. FOCUS

Chunlu Li et. al [32] megközelítése a korábbi munkákhoz képest azért különleges mert, a 3D-s arcok rekonstrukcióját és az okklúziók szegmentálását együttesen végzi. Ráadásul, az általuk fejlesztett pipeline csak gyenge felügyelet mellett tanul, és nincsen szüksége annotációkra a különböző típusú okklúziókhoz. Ezenkivül, az általuk használt arc autoencoder lehetővé teszi az arcmodell hatékonyabb illesztését. Ahhoz, hogy növelni tudják a megvilágítással és más tényezőkkel szembeni robosztusságot, implementáltak egy úgynevezett perceptuális költségfüggvényt, amellyel a szegmentáló hálózat képes a szemantikus jellemzők helyett csak a független pixeleken keresztül gondolkodni.

A fenti ábrán láthatjuk a hálózat felépítését, de hogyan is mükődik? [32] Adott egy IT bemeneti kép, a rekonstrukciós hálózat, R, megbecsüli a látens paramétereket és ezt követően egy olyan IR képet állít elő, amely csak az arcot tartalmazza. Ezután az IT és IR képeket betápláljuk az S szegmentáló hálózatba, amely megjósolja az M szegmentációs maszkot. A szaggatott vonalak azt mutatják, hogy M -et arra használják, hogy az IT és IR képekben a becsült okklúziókat kitakarják, ezáltal összerakott okklúzió mentes képeket kapnak, nevezetesen IT M és IR M . Megfigyelhetjük, hogy Chunlu Li et. al nagy hangsúlyt fektettek bele a robosztus arc rekonstrukcióra okklúziók mellet. A munkánk szempontjából azért fontos, mert az okklúziók mindenütt jelen vannak. Ezért is választottuk a F OCU S-t, mert nekik sikerült egy speciális modellt implementálni, ami képes kezelni az okklúziókat, anélkül, hogy szükséges lenne nagy mennyiség″u adatra, vagy olyan erőforrásra,



3. Ábra. Chunlu Li et. al javasolt megközelítés pipeline-ja. Forrás:[33]

amihez nincs hozzáférésünk. A továbbiakban részletesebben megvizsgáljuk a F OCU S építőelemeit és együttm˝ukődésüket.

[32] Ahogy, már korábban is meg volt említve, a F OCU S célja egy 3D-s arc rekonstrukciója egyetlen képből, súlyos okklúziók esetén is. Ezen kihívást jelentő probléma megoldásához egy modellalapú arc autoencoder-t, R-t, és egy szegmentáló hálózatot, S-t implementáltak, ahogyan az a fenti ábrán is szemléltetve van.

Az arc rekonstrukciójához a szegmentációs maszk a modellillesztés során kivágja a becsült okklúziókat, így a rekonstrukciós hálózatot robosztussá teszi az okklúziókkal szemben. A szegmentáláshoz a rekonstruált eredmény referenciaként szolgál, növelve a szegmentálás pontosságát.

Ebben a szakaszban megvizsgáljuk hogyan m˝uködik a két hálózat, továbbá, hogyan kapcsolódnak egymáshoz, és milyen előnyöket nyújtanak egymás számára.

2.2.1. Modellalapú autoencoder

[32] A modellalapú arc autoencoder, R, várhatóan rekonstruálja a teljes arc megjelenését a látható arctartományokból a képen, IT -n. Ez egy kódolóból ,grafikus renderelőből, és egy dekóderből áll. A kódoló megbecsüli a látens parmétereket $\Theta = [\ ,\ ,\ ,\ c]$ R257, azaz a 3D alak R144, a 3DMM textúrája R80, a megvilágítás R27, és a jelenet kamera paraméterei c R6. Adott látens parméterekkel a dekóder a bemeneti képen látható arc arcképét állítja elő $IR = R(\Theta)$. Majd ehhez egy olyan felügyelet nélküli szegmentáló hálózatot vezetnek be, melynek kimenetelét a modellillesztés során az okklúziók edfedésére lehet használni, és így az autoencoder-t robosztussá teszi az okklúziókkal szemben.

2.2.2. Szegmentációs hálózat

[32]A szegmentáló hálózat, S, veszi az IT képet és a szintetizált képet, IR-t, bemenetként, és megjósolja a bináris maszkot, $M = S(IT\ , IR)$, annak leírására, hogy egy pixel az arcot ábrázolja-e (1) vagy nem (0). Mivel az IR tartalmazza a becsült arcot, előzetes tudást biztosít a szegmentáló hálózatnak ´es segíti a becslést.

Az arc autoencoder és a szegmentáló hálózat képzés során összevannak kapcsolva, hogy egy szinergikus hatást váltson ki, ami a szegmentálást pontosabbá teszi és a rekonstrukciót robosztussabbá teszi okklúziók jelenlétében.

2.2.3. Tanítás

[32] Az arc autoencoder és a szegmentáló hálózat kölcsönös függőségei mi- att egy $Expectation - M\ aximization(EM\ )$ típusú stratégiát alkalmaztak, ahol a két hálózatot váltakozva tanították be. Ez lehetővé tette a stabil konvergenciát a betanítási folyamat során. Mint más EM típusú tanítási stratégiákhoz hasonlóan, a tanítási folyamatjuk a modell paraméterinek durva inicializálásával kezdődik, amelyet felügyelet nélküli módon kaptak meg.

A szegmentáló hálózat képzésekor az arc autoencoder paraméterei rögzítettek, és csak a szegmentáló hálózatot optimalizáljuk. Az an- notált adatok keresése helyeken, négy költségfüggvényt javasoltak, amelyek kikényszerítik a képek közötti hansolóságokat. A költségfüggvények dolga hogy, eldöntse egy adott pixelről hogy az arc része vagy ellenkezője. Ezek vagy perceptuális szinten vagy pixel szinten dolgoznak, hogy teljes mértékben kihasználják

a vizuális nyomokat. Részletesebben erről a szerzők munkájában [32] lehet olvasni.

Betanítás során, úgy van irányitva a szegmentáló hálózat, hogy egyensúlyt keressen, az olyan képpontok elvetése között, amelyeket az autoencoder nem tud jól értelmezni, és az olyan képpontok megőrzése között amelyek fontosak a bemeneti kép és az előállított arc perceptuális reprezentációjának megőrzése´erdekében. Ezáltal nincsen szükség az okklúziók felügyeletére.

Az autoencoder betanítása során, tovább optimalizálták az autoencoder paramétereit, közben a szegmentáló hálózat rögzítve van. Az autoencoderhez tartóznak az alábbi költségfüggvények[32]:

$Lpixel = (ITIR)M22(4)Lper = cos(F(IT), F(IR))(5)Llm = lmTlmR22(6)$

2.2.4. Adathalmazok

[32]Az általuk felhasznált adatbázisok a CelebA-HQ és az AR adatbázisok. Ezek segítségével értékelik az illesztés és az arc szegmentlás hatékonyságát. Illetve alakrekonstrukció pontosságát is lemérték a NoW adatbázis részhalmazain.

2.3. Értékelés

Mint láthatjuk a fent említett megközelítések mind kiváló eredményeket nyújtanak, de megfigyelhetjük hogy a két munkának a szerzői teljesen különböző motivációval rendelkeztek amikor a megközelítéseiken dolgoztak. A DECA[31] főbb motivációja az emberi arc részleteinek megőrzése, a FOCUS[32] célja pedig egy gyors modell kialakítása volt, amely korábbi munkákhoz képest jobban kezelte az okklúziókat. A mi célunk, egy olyan modell kialakítása amely képes megőrízni a részleteket, de egyben legyen képes konzisztens működést nyújtani az okklúziók jelenlétében is

Tehát a tervezett pipeline, amit a későbbiekben részletesebben megvizsgálunk, kettő egymást támogató hálózatból áll. Egyik fő épitőeleme a pipelinenak a DECA[31] alapján készített rekonstrukciós hálózat. A másik, a FOCUS[32] megközelítésében implementált szegmentáló hálózat. E két hálózaton majd egy EM típusú stratégiát alkalmazunk a tanítási folyamat során.

Rank	Method	Median(mm)	Mean(mm)	Std(mm)
1.	**FOCUS (Ours)**	**1.04**	**1.30**	**1.10**
2.	DECA[Feng et al., SIGGRAPH 2021]	1.09	1.38	1.18
3.	Deep3DFace PyTorch [Deng et al., CVPRW 2019]	1.11	1.41	1.21
4.	RingNet [Sanyal et al., CVPR 2019]	1.21	1.53	1.31
5.	Deep3DFace [Deng et al., CVPRW 2019]	1.23	1.54	1.29
6.	3DDFA-V2 [Guo et al., ECCV 2020]	1.23	1.57	1.39
7.	MGCNet [Shang et al., ECCV 2020]	1.31	1.87	2.63
8.	PRNet [Feng et al., ECCV 2018]	1.50	1.98	1.88
9.	3DMM-CNN [Tran et al., CVPR 2017]	1.84	2.33	2.05

4. Ábra. NoW Challenge benchmark eredményei Forrás: https://github.com/unibas-gravis/Occlusion-Robust-MoFA

3. 3D arcmodell konstruálása

[5] A 3D-s arcok legelterjedtebb statisztikai modellje a 3D Morphable Models (3DMM), amelyet Blanz és Vetter [8] mutatott be a közösségnek. *Bernhard et al.* [9] munkája alapján a 3D Morphable Face Model egy generatív modell az arcformára és a megjelenés modellje, amely két kulcsfontosságú ötleten alapul:

Először is, minden arc sűrű pont-pont megfeleltetésben van, amelyet általában egy regisztrációs eljárás során egy sor példaarcon állítanak elő, majd a további feldolgozási lépések során is megmarad.Ennek a megfeleltetésnek köszönhetően az arcok lineáris kombinációi definiálhatók egy értelmes módon, morfológiailag valósághű arcokat (morfokat) létre hozva.

A második ötlet az arc alakjának és színének szétválasztása, és ezek függetlenítése a külső tényezőktől, például a megvilágítástól és a kamera paraméterektől.

A *morphable* modell magában foglalhat egy statisztikai modellt az arcok eloszlásáról, amely egy főkomponens elemzés az eredeti munkában [8], majd más tanulási technikákat is tartalmazott a későbbi munkák során.

3.1. Arcképfelvétel

Minden 3DMM legfontosabb összetevője a 3D alakzatok reprezentatív készlete, általában a megfelelő megjelenési adatokkal együtt. A mintakészlet létrehozásának tipikus módja, hogy az adatokat a valós világból kapjuk meg. Ebben a szakaszban rövid áttekintést adunk a különböző megközelítésekről, amelyeket az arcadatok, valamint az arcképek adatainak megszerzésére használatosak [9].

A bemeneti adathalmazok létrehozása a 3DMM-ek számára kontrollált körülmények között történő felvételre korlátozódnak, szemben a nagyobb kihívást jelentő, ”in-the-wild” felvételekkel.

Megjegyzendő, hogy a kontrollált 3D arcfelvétel nem mindig szükséges. Voltak kísérletek arra, hogy a 3DMM-eket közvetlenül képekből tanulják meg. Például Cashman és Fitzgibbon munkája 2012-ben [10] és a legmodernebb mélytanuláson (*deep learning*) alapuló módszerek egyszerre tanulnak 3DMM-et és regresszió-alapú illesztést 2D-s képzési adatokból.

3.2. Alakfelvétel

A háromdimenziós forma vitathatatlanul a 3DMM legfontosabb összetevője [9]. Az alakzat ábrázolásának kérdése a 3DMM-ek összefüggésében nem került széles körben figyelembe vételre.

Messze a leggyakrabban használt reprezentáció a háromszögháló. Vannak ritka kivételek, de ebben a munkában ezeket nem fogjuk megemlíteni.

A háromszöghálós reprezentáció sűrű megfeleltetéssel megköveteli, hogy minden minta azonos topológiát mutasson, és hogy a csúcsok minden mintán ugyanazt a szemantikai pontot kódolják. A minták közötti megfelelés megállapítása önmagában is kihívást jelentő téma. Ebben a szakaszban a nyers 3D adatokra összpontosítunk.

3.2.1. Statisztika alapú modell illesztési módszerek

A statisztikai 3D arcmodell [5] a legnépszerűbb módszer az előzetes információ hozzáadására, mivel ezek kódolják az arc geometriai variációit, esetleg a megjelenéssel együtt.

Az ilyen modellek tartalmaznak egy átlagos arcot, valamint annak geometriájának és az arcmintázatának a variációit és a megjelenését.

3D arcmodell illesztése fényképre a modell paraméterein kívül, a 3D póz és a megvilágítás meghatározásával történik úgy, hogy az eredményül kapott 3D-s arc képsíkjába történő vetülete a lehető legjobban hasonlítson az adott képhez.

3.2.2. Geometriai módszerekhez

A geometriai módszerek [9] közvetlenül becslik egy alakzat 3D-s koordinátáit vagy ugyanazon felület megfigyelésével két vagy több nézőpontból (ebben az esetben a kihívás a megfelelő pontok azonosítása a képek között), vagy pedig egy vetített minta megfigyelésével (ebben az esetben a kihívás az ismert minta és a róla készült vetület kép közötti megfeleltetés azonosítása).

A módszerek vagy aktívnak tekinthetők, azaz fényt vagy más jeleket sugároznak, vagy passzívak.

A lézerszkennerek, a *Time-of-Flight* érzékelők és a strukturált fényrendszerek *aktív* rendszerek, míg a több nézetből álló fotogrammetria *passzív* alternatíva.

3.2.3. Fotometrikus módszerek

A fotometrikus módszerek [9] jellemzően a felület orientációját becslik, amelyből integrálással vissza lehet nyerni a 3D alakot.

A kihívás itt az, hogy olyan modelleket válasszunk, amelyek pontosan megragadják a felszín reflexiós tulajdonságait, valamint elegendő mérési eredményt kapjunk ahhoz, hogy e modellek invertálása jól megoldható legyen.

A geometriai módszerekhez képest a fotometriai módszerek jellemzően nagyobb alaki részletességet kínálnak, és nem függnek a összeilleszthető jellemzők meglététől (tehát sima, jellegtelen felületek esetén is alkalmazhatóak), de gyakran szenvednek alacsony frekvenciájú torzításoktól a rekonstruált képekben a fényvisszaverő képesség és a megvilágítás modellezési hibái miatt.

A fotometrikus sztereó [11] a felszín normálisa adja meg minden egyes képpontnál a jelenet rögzített pozícióból történő megfigyelésével, legalább három különböző megvilágítási körülmény között.

A szükséges képkockák számát spektrális multiplexálással lehet csökkenteni [12].

3.2.4. Hibrid módszerek

A hibrid módszerek [9] a geometriai és a fotometriai módszerek kombinációja.

Csökkentik a a fotometriai módszereknél jellemzően jelenlévő alacsony frekvenciájú torzítást, és növelik a nagyfrekvenciájú részleteket a geometriai módszerekhez képest.

Diego Nehab *et al.* [13] javaslata egy olyan módszer, amely helymeghatározó információk alacsony frekvenciáját egyesíti a felszíni normálisok magas frekvenciájával. A módszer különösen hatékony, mivel csak egy lineáris egyenletrendszer megoldása szükséges.

4. Neurális hálózatok

4.1. Mi az a neurális hálózat?

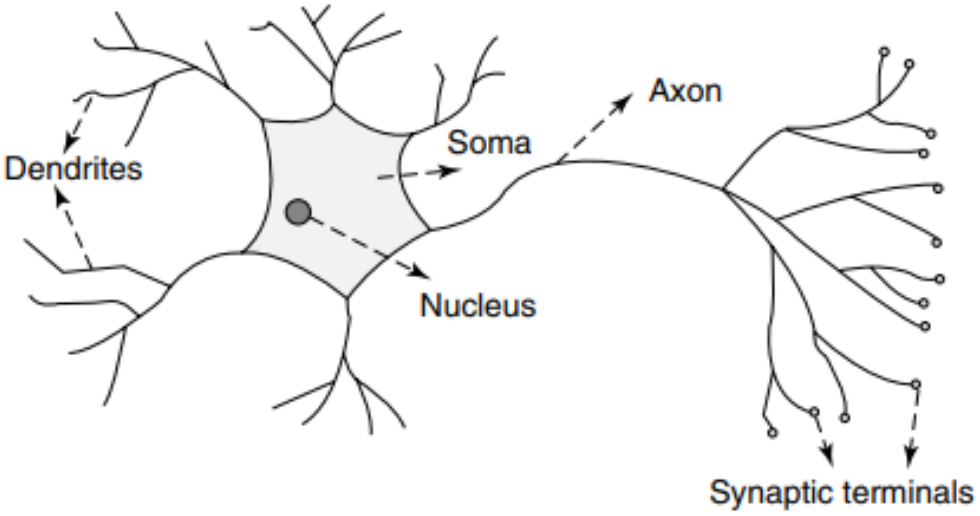
Az emberi agy bizonyítja a hatalmas neurális hálózatok [15] létezését, amelyek sikeresen végeznek el kognitív, észlelési és irányítási feladatokat. Ilyen számításigényes feladat például az arcfelismerés, a beszéd és a testmozgás. Az agy hatékonyan kihasználja a masszív párhuzamosságot, számítási struk-

túrája nagymértékben párhuzamos, és jó információfeldolgozási képességgel rendelkezik.

Az emberi agy több, mint 10 milliárd egymással összefüggő neuron gyűjteménye. Mindegyik neuron egy sejt [1. ábra], amely biokémiai reakciókat használ az információ fogadáshoz, feldolgozáshoz és továbbításhoz.

Az idegrostok faszerű hálózatai, az úgynevezett dendritek kapcsolódnak a sejttesthez, ahol a sejtmag található. A sejttestből egyetlen hosszú rost nyúlik, melyet *axon*nak neveznek. Az *axon* szálakra és alszálakra ágazik, majd szinapszisain keresztül kapcsolódik más neuronokhoz. Az itt létrejövő szinaptikus kapcsolat erőssége határozza meg az emberi agy tanulását[15] .

A jelek átvitele egyik neuronról a másikra a szinapszisoknál egy összetett kémiai folyamat, amelyben specifikus közvetítő anyagok szabadulnak fel a kapcsolódási pont küldői oldalán. A folyamat hatására a fogadó sejtben emelkedik vagy csökken az elektromos feszültség[15].



5. Ábra. Neuron felépítése. Forrás:[15]

4.2. Mesterséges neurális hálózatok

A mesterséges neurális hálózatok (ANN-artificial neural networks) az előző fejezetben [subsection 4.1] említett fejlettebb élő szervezetek agyát alkotó neuronokról kapták nevüket.

Mint azt *Abraham* [15] is említi, a neurális hálózatok alapvető feldolgozási elemeit mesterséges neuronoknak nevezzük, vagy csak egyszerűen neuronoknak vagy csomópontoknak (*node*). A neuron leegyszerűsített matematikai modelljében a szinapszisok hatásai kapcsolati súlyokkal vannak reprezentálva, amelyek szabályozzák a bemeneti jelek hatását.

A neuronok által mutatott non-lineáris karakterisztikát átviteli függvény segítségével ábrázoljuk. A neuron impulzusát ezután a bemeneti jelek súlyozott összegeként számíthatjuk ki az átviteli függvénnyel transzformálva.

Egy mesterséges neuron tanulási képessége a súlyok megfelelő beállításával érhető el egy választott tanító algoritmus felhasználásával.

Egy tipikus mesterséges neuron és egy többrétegű neurális hálózat modellezése a 2. ábrán látható. A 2. ábra szerint, ahogy a nyilak is mutatják, az $x_1, ..., x_n$ bemenetekről érkező jeláramlás egyirányúnak tekinthető, csakúgy mint a neuron kimeneti jelfolyama. A neuron kimeneti jelét O a következő összefüggés adja:

$$O = f(net) = f(\sum j = 1^n wjxj) \tag{1}$$

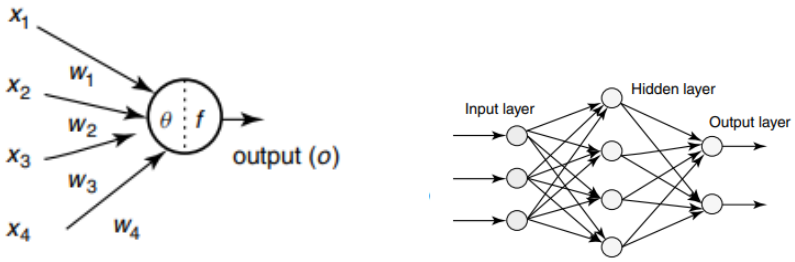
ahol wj a súly vektor és $f(net)$ az *aktivációs* (átviteli) függvény. A *net* változó a súly és a bementi vektorok skaláris szorzataként definiálható,

$$net = w^T x = w1x1 + ... + wnxn \tag{2}$$

ahol T egy mátrix transzponálását jelöli, és a legegyszerűbb esetben a kimeneti érték O kiszámítható, mint

$$O = f(net) = \Big\{1, ; ha; w^T x; \geq; \theta \ 0, \text{különben} \tag{3}$$

ahol θ-t *küszöbszintnek* (threshold) nevezzük. Ezt a típusú aktivációs függvényt *küszöb aktivációs függvénynek* nevezzük.

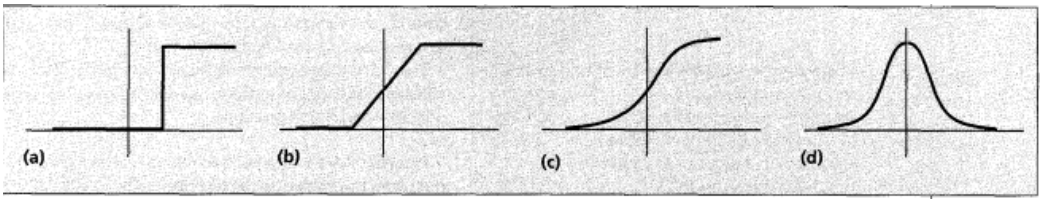


(a) Mesterséges neuron (b) Többrétegű neurális háló

6. Ábra. Mesterséges neuron felépítése és egy többrétegű neurális háló
Forrás: [15]

A *küszöbfüggvényhez* hasonlóan más aktivációs függvényekkel is előállítható az adott neuron bemenetre kapott válasz. Ilyen függvények a

- lineáris,
- szigmoid,
- tangens hiperbolikusz.



7. Ábra. (a) küszöbfüggvény, (b) lineáris, (c) szigmoid, (d) Gauss
Forrás:[16]

Ma az egyik leggyakrabban használt aktivációs függvény a szigmoid [18], melyet a következő képlettel határoznak meg:

$$y = \frac{1}{1 + e^{-(a-\theta)b}} \tag{4}$$

ahol a az aktiválás, b pedig a görbe alakját szabályozza.

4.3. Neurális hálózat architektúra

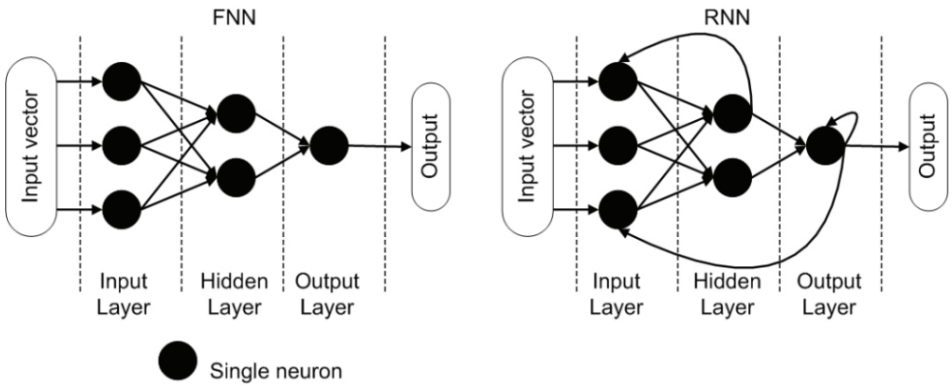
Bár a mesterséges neuron működési elvei és egyszerű szabályrendszere elsőre talán nem tűnik érdekesnek, azonban e modellek teljes potenciálja és számítási teljesítménye akkor kel életre, amikor mesterséges neurális hálózatokká kezdjük őket összekapcsolni [2. ábra]. Ezek a mesterséges neurális hálózatok kihasználják az egyszerű tényt, hogy a komplexitás néhány alapvető szabályból tud növekedni.

A mesterséges neurális hálózatok képesek komplex, valós problémák megoldására azáltal , hogy alapvető építőelemeikben (*mesterséges neuronok*) feldolgozzák az információt nemlineáris, elosztott, párhuzamos és lokális módon.

Ahogy Krenker *et al.* [19] is leírta, az egyes mesterséges neuronok összekapcsolásának módját *topológiának*, *architektúrának* vagy *gráfnak* nevezzük. A tény, hogy az összekapcsolás számos lehetséges módon történhet több alkalmazható topológiát eredményez, amelyek két fő csoportra bonthatók.

A 4. ábra ezt a két topológiát mutatja. Az ábra bal oldala egy egyszerű feedforward topológiát ábrázol, ahol az információ a bemenetekről kimenetekre áramlik egyirányúan. Az ábra jobb oldalán pedig egy egyszerű rekurzív (*recurrent*) topológiát ábrázol, ahol az információ egy része nem csak egy irányban áramlik a bemenetről a kimenetre, hanem ellenkező irányban is.

Fontos megemlíteni, hogy a mesterséges neurális hálózat könnyebb kezelése és matematikai leírása érdekében az egyes neuronokat rétegekbe soroljuk. A 4. ábrán láthatjuk a bemeneti, a rejtett és a kimeneti réteget.



8. Ábra. Feed-forward (FNN) és recurrent (RNN) neurális hálózati topológiák. Forrás:[19]

A háló input rétegében minden neuron kapcsolatban áll a *rejtett* (köztes) réteggel, így tovább adhatja a bemenetként kapott adatokat. A bemeneti réteg neuronjai súlyozott szinapszisokkal kapcsolódnak a belső rétegekhez.

A mesterséges neurális hálózat topológiájának kiválasztásával és felépítésével még csak a feladataink felét fejeztük be mielőtt a hálót az adott probléma megoldására használhatnánk. Csakúgy, mint a biológiai neurális hálózatoknak is meg kell tanulniuk a megfelelő válaszokat különböző környezeti bemenetekre, a mesterséges neurális hálózatoknak is pontosan ezt kell tenniük.

A következő lépés tehát, hogy *betanítsuk* a mesterséges neurális hálózatot a helyes válaszokra. Erre négy lehetőségünk van:

- a *supervised* (felügyelt)
- a(z) *un-supervised* (felügyelet nélküli)
- a *reinforcement* (megerősítéses)
- és a hibrid

tanulás.

Függetlenül attól, hogy melyik módszert választjuk, a tanulás feladata, hogy a tanulási adatok alapján beállítsuk a súlyok és az előfeszítések értékeit, hogy minimalizáljuk a költségfüggvényt.

4.4. Neurális hálók tanítása

A tanulási folyamat [16] az ANN kontextusában a hálózat architektúrája és a kapcsolati súlyok frissítéseként értelmezhető annak érdekében, hogy a háló hatékonyan tudjon elvégezni egy adott feladatot. A hálózatnak a rendelkezésre álló tanító mintákból kell megtanulnia a kapcsolati súlyokat. A teljesítmény idővel javul a súlyok iteratív frissítésével. Az ANN-eket automatikus, példákból való tanulása teszi vonzóvá és érdekessé. Szakértők által definiált szabályok helyett az ANN-ek a mögöttes szabályokat tanulják meg (pl. bemeneti-kimeneti kapcsolatokat) a megadott reprezentatív példák gyűjteményéből.

4.4.1. Felügyelt tanulás (*supervised learning*)

A felügyelt tanulás [19] egy olyan gépi tanulási (*machine learning*) technika, amely egy mesterséges neurális hálózat paramétereit tanítási adatokból állítja be. A tanuló ANN feladata, hogy beállítsa paramétereinek értékét bármely érvényes bemeneti értékre, miután látta a kimeneti értéket. A képzési adatok a bementi és a kívánt kimeneti értékek párjaiból állnak, amelyeket adatvektorokban ábrázolnak.

A felügyelt tanulást *klasszifikációnak* (*classification*) is nevezhetjük, ahol *osztályozók* (*classifiers*) széles skálája áll rendelkezésünkre, amelyek mindegyikének megvannak az erősségei és gyengeségei. A megfelelő osztályozó

(többrétegű perceptron, k-nearest neighbour algoritmus, Gauss-keverék modell, stb..) egy adott problémára való kiválasztása azonban inkább művészet, mint tudomány. A felügyelt tanulás egy adott problémájának megoldásához különböző lépéseket kell figyelembe venni [19].

Az első lépésben meg kell határoznunk a tanítási minták típusát. A második lépésben olyan képzési adathalmazt kell gyűjtenünk, amely kielégítően leírja az adott problémát. A harmadik lépésben az összegyűjtött képzési adathalmazt olyan formában kell leírnunk, amely érthető a kiválasztott ANN számára. A negyedik lépésben elvégezzük a tanulást, és a tanulás után tesztelhetjük a tanult ANN teljesítményét a teszt (validációs) adathalmazzal. A validációs adathalmaz olyan adatokból áll, amelyeket a tanulás során nem vezettünk be a mesterséges neurális hálózatba.

4.4.2. Felügyelet nélküli tanulás (*unsupervised learning*)

A felügyelet nélküli tanulás [19] egy olyan gépi tanulási (*machine learning*) technika, amely egy ANN paramétereit megadott adatok és egy minimalizálandó költségfüggvény alapján állítja be. A költségfüggvény bármilyen függvény lehet, amit a feladat határoz meg.

Felügyelet nélküli tanulást leginkább olyan alkalmazásokban használnak, amelyek a becslési problémák körébe tartoznak, mint például a statisztikai modellezés, filterezés és a klaszterezés. A felügyelet nélküli tanulás során arra törekszünk, hogy meghatározzuk, hogyan szerveződnek az adatok. Ez a felügyelt és a megerősítéses tanulástól abban különbözik, hogy az ANN csak *címkézetlen* (*unlabeled*) mintákat kap. A felügyelet nélküli tanulás egyik gyakori formája a klaszterezés, ahol az adatokat hasonlóságuk alapján próbáljuk különböző klaszterekbe sorolni.

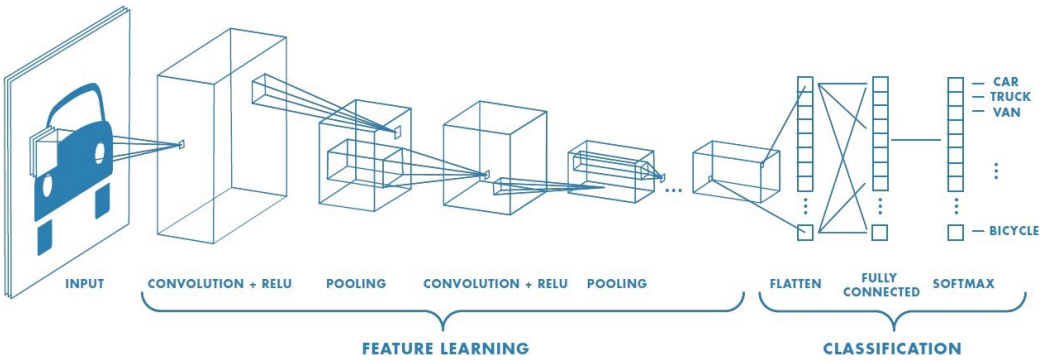
4.4.3. Megerősítéses tanulás (*reinforcement learning*)

A megerősítéses tanulás [19] egy olyan gépi tanulási (*machine learning*) technika, amely egy olyan mesterséges neurális hálózat paramétereit állítja be, ahol általában az adatok nincsenek előre megadva, hanem a környezettel való kölcsönhatásokból generálódnak. A megerősítéses tanulás azzal foglalkozik, hogy egy mesterséges neurális hálózatnak hogyan kellene viselkednie egy adott környezetben annak érdekében, hogy maximalizálja a hosszú távú eredményeket.

Miután a maximalizálandó visszatérési függvényt meghatároztuk, a megerősítéses tanulás számos algoritmust használ a maximális visszatérést eredményező szabályok (*policy*) meghatározására. Az első lépésben egy naiv *brute force* algoritmus kiszámítja a *visszatérési függvényt* (*return function*) minden lehetséges szabályhoz, és kiválasztja azt, amelyik a legnagyobb visszatéréssel rendelkezik. Ennek az algoritmusnak nyilvánvaló gyengesége a rendkívül magas vagy akár végtelen számú lehetséges szabály esetében rejlik. Ez a gyengeség kiküszöbölhető érték függvényes megközelítésekkel vagy közvetlen szabály becslésekkel. Az értékfüggvényes megközelítések megpróbálnak egy olyan szabályrendszert találni, amely maximalizálja a visszatérést.

Ezek a módszerek konvergálnak a helyes becslésekhez egy rögzített szabály esetén, és az optimális szabály megtalálására is használhatóak. Az értékfüggvényes megközelítéshez hasonlóan a közvetlen szabály becslés is képes megtalálni az optimális szabályt.

4.5. Konvolúciós neurális hálózatok



9. Ábra. CNN(Convolutinal Neural Network)

A konvolúciós neurális hálózat (CNN) [21] egy mély tanulási (*deep learning*) megközelítés, amelyet széles körben használnak komplex problémák megoldására és felülmúlja a hagyományos gépi tanulási megközelítéseket.

A konvolúciós neurális hálózat (CNN), gyakran ConvNet-nek is nevezik, mély feed-forward architektúrával rendelkezik, és jobban képes általánosítani, mint a teljesen összekapcsolt rétegekkel rendelkező hálózatok.

A CNN, mint a hierarchikus jellemző detektorok biológiailag inspirált koncepciója, képes nagymértékben megtanulni absztrakt jellemzőket, és ha-

tékonyan képes azonosítani az objektumokat.

A CNN-t az alábbiak miatt tartják jobbnak más klasszikus modellekhez viszonyítva:

Először is, a CNN-ek alkalmazásának legfőbb előnye a súlymegosztás koncepciójában rejlik , amelynek köszönhetően a betanítandó paraméterek száma jelentősen csökken, ami jobb általánosításhoz vezet. A kevesebb paraméter miatt a CNN egyszerűen betanítható, és nem szenved túlillesztéstől(overfitting).

Másodszor, az osztályozási szakasz beépül a jellemző kinyerési szakaszba, mindkettő használ tanulási folyamatot.

Harmadszor, a mesterséges neurális hálózat (ANN) általános modelljeinek felhasználásával nagy hálózatokat sokkal nehezebb megvalósítani, mint CNN-ben.

A CNN-eket széles körben használják különböző területeken figyelemreméltó teljesítményüknek köszönhetően, mint például a képosztályozás, a tárgyak felismerése, az arcfelismerés, a beszéd felismerés, járműfelismerés stb.

4.6. A konvolúciós neurális hálózat általános modellje

4.6.1. Általános modell

Az ANN általános modellje [21] egyetlen bemeneti és kimeneti réteggel, valamint több rejtett réteggel rendelkezik.

Egy bizonyos neuron fogadja az X bemeneti vektort, és Y kimenetet állít elő azáltal, hogy valamilyen F függvényt hajt végre rajta az alábbi általános egyenlet alapján (5).

$$F(X, W) = Y \tag{5}$$

,ahol W a súlyvektor, amely a két neuron közötti összeköttetés erősségét jelöli két szomszédos réteg között.

A kapott súlyvektor most már felhasználható a képosztályozáshoz.

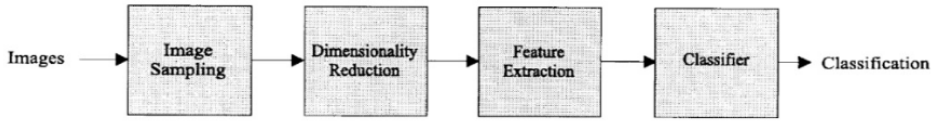
Jelentős mennyiségű irodalom létezik a képek pixelalapú osztályozásával kapcsolatban, azonban az olyan kontextuális információk, mint a kép alakja vagy a kép formája jobb eredményt adnak.

A CNN a kontextuális információkon alapuló osztályozási képessége miatt kap egyre nagyobb figyelmet.

A CNN általános modellje négy komponensből áll, nevezetesen

- konvolúciós réteg
- pooling réteg
- aktivációs függvény
- teljesen összekapcsolt réteg

Az egyes komponensek működését az alábbi ábra szemlélteti.



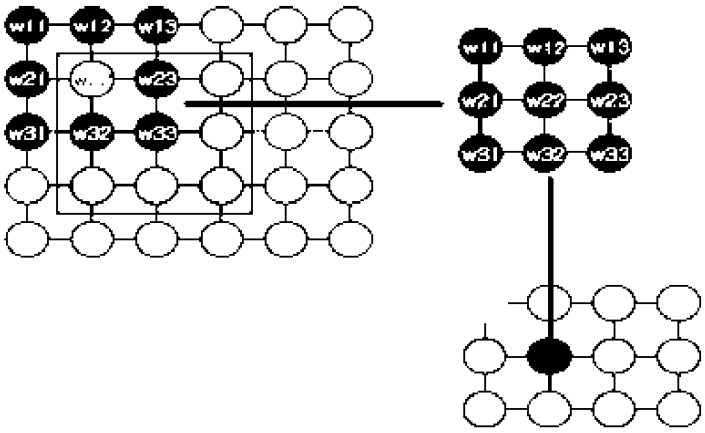
10. Ábra. [21] A CNN elemi összetevői

4.6.2. Konvolúciós réteg

[21]Az osztályozandó képet a bemeneti rétegnek adjuk meg, a kimenet pedig az előre megjósolt osztálycímke, amelyet a képből kinyert jellemzők alapján számítunk ki.

A következő rétegben lévő egyes neuronok az azt követő rétegben lévő neuronokhoz kapcsolódnak. Ezt a helyi korrelációt receptív mezőnek nevezzük.

A bemeneti kép lokális jellemzőit a receptív mező segítségével nyerjük ki. Az előző rétegben egy adott régióhoz tartozó neuron receptív mezeje egy súlyvektort alkot, amely a sík minden pontján egyenlő marad, ahol a sík a következő rétegben lévő neuronokra utal. Mivel a síkban lévő neuronok azonos súlyokkal rendelkeznek, így a különböző helyeken előforduló hasonló jellemzők a bemeneti adatokon belül felismerhetők.



11. Ábra. [21] Az adott neuron receptív mezeje a következő rétegben

A súlyvektor, más néven szűrő vagy kernel, a bemeneti vektoron csúszik át a *featuremap* létrehozásához. A szűrő vízszintes és függőleges irányú csúsztatásának módszerét nevezzük konvolúciós műveletnek. A helyi receptív mező jelensége miatt a betanítható paraméterek száma jelentősen csökken.

A következő rétegben az (i,j) helyhez tartozó A_{ij} kimenet konvolúciós művelet alkalmazása után kerül kiszámításra az alábbi képlet segítségével:

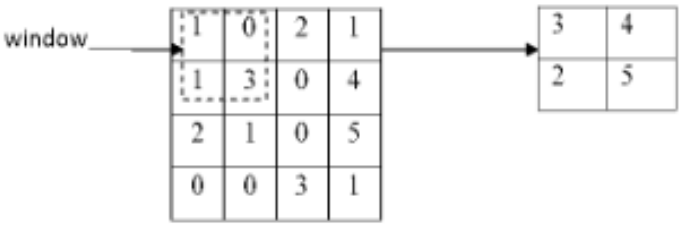
$$a_{ij} = \sigma((W * X)_{ij} + b) \tag{6}$$

, ahol X a rétegnek adott bemenet, W a bemeneten áthaladó szűrő vagy kernel, b az eltolás, * a a konvolúciós műveletet, és σ a hálózatba bevezetett non-linearitást jelöli.

4.6.3. Összevonó réteg

A konvolúciós réteget összevonó (pooling) vagy almintavételező (sub-sampling) [21] réteg követi. A *pooling* technika használatának fő előnye, hogy jelentősen csökkenti a betanítható paraméterek számát, és bevezeti a fordítási invarianciát.

Az összevonás (*pooling*) művelet elvégzéséhez kiválasztunk egy ablakot, és az abban az ablakban lévő bemeneti elemeket átadjuk egy összevonó (*pooling*) függvénynek. a [8. ábrán] látható módon.



12. Ábra. [21] 2 x 2 ablak kiválasztásával végzett összevonási művelet

A pooling függvény egy másik kimeneti vektort generál.

Létezik néhány összevonási technika, mint például az átlagos összevonás (*average pooling*). és a *max-pooling*, amelyek közül a *max-pooling* a leggyakrabban használt módszer, mely jelentősen csökkenti a leképezés méretét.

A hibák kiszámítása során a hiba nem terjed vissza a győztes egységre.

4.6.4. Teljesen összekapcsolt réteg

A teljesen összekapcsolt réteg [21] hasonló a hagyományos modellek teljesen összekapcsolt hálózatához.

Az első fázis kimenete (a konvolúciót és a összevonást ismétlődően tartalmazza) a teljesen összekapcsolt rétegbe kerül, és a súlyvektor és bemeneti vektor pontproduktumát számoljuk ki a végeredmény kiszámítása érdekében.

A *gradiens süllyedés*, más néven kötegelt módú tanulás vagy offline algoritmus, csökkenti a költségfüggvényt a költség becslésével egy teljes tanítói adathalmaz felett, és a paramétereket csak egy korszak után frissíti, ahol egy korszak a teljes adathalmaz átfutásának felel meg. Ez globális minimumokat eredményez, de ha a képzési adathalmaz mérete nagy, akkor a hálózat tanításához szükséges idő jelentősen megnő. A költségfüggvény
csökkentésének ezt a megközelítését felváltotta a *sztochasztikus gradiens süllyedés*.

4.6.5. Aktivációs függvény

Számos szakirodalom létezik, amely a hagyományos gépi tanulási algoritmusokban szigmoid aktivációs függvényt használ. A nemlinearitás bevezetése érdekében a Rectified Linear Unit (ReLU) [21] használata jobbnak bizonyult az előbbinél két fő tényező miatt.

Először is, a ReLU parciális deriváltjának kiszámítása egyszerű. Másodszor, miközben figyelembe vesszük a képzési időt mint az egyik tényezőt, a telítődő nemlinearitások, mint a szigmoid lassabbak, mint a nem telítődő nemlinearitások, mint a ReLU. Harmadszor, ReLU nem engedi, hogy a gradiensek eltűnjenek, de a ReLU hatékonysága romlik, ha nagy gradiens áramlik át a hálózaton, és a súly frissítése miatt a neuron nem aktiválódik, ami a *Dying ReLU* problémához vezet, amely egy jelentős probléma.

Ez a probléma megoldható a *Leaky ReLU* segítségével, ha $x > 0$, a függvény aktiválódik. mint $f(x) = x$, ha pedig $x < 0$, akkor a függvény αx-ként aktiválódik, ahol α egy kis konstans.

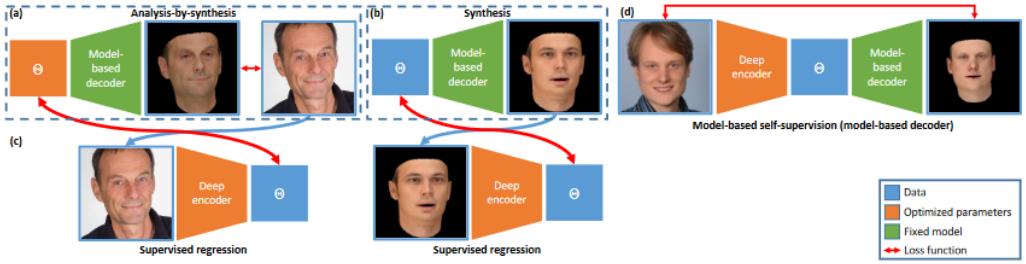
5. Deep Learning módszerek

Az előző szakaszok [3D arcmodell konstruálása] 2D-ből 3D arc rekonstrukciós módszerei *modelleket* használnak az előzetes tudás megtestesítésére [5]: a statisztikai modellillesztési módszerek tartalmaznak egy geometriai (és általában textúra) modellt, a fotometriai módszerek pedig a az arc fényvisszaverő képességét.

Ezzel szemben a mély tanulási módszerek közvetlenül tanulják meg a 2D kép és a 3D arc közötti leképezést, az előzetes ismeretek felhasználásával a betanított hálózatokban.

Több indokból is előnyösebb mély tanulási módszerekkel dolgozni.[9]A modellezési oldalon a nemlineáris, mély reprezentációk használata lehetőséget nyújt arra, hogy felülmúljuk a klasszikus lineáris vagy multi-lineáris modelleket az általánosítás szempontjából, tömörség és specifikusság tekintetében [4].

A paraméter becslési oldalon ki tudjuk használni a mély hálózatok előnyeit, a gyorsaságot és a robusztusságot, hogy megbízható teljesítményt érjünk el a nem ellenőrzött képeken.



13. Ábra. A klasszikus analízis-szintézis és a mélytanulási megközelítések közötti kapcsolat. Forrás:[9]

5.1. Deep Face modellek

A hagyományos modellezési technikák célja, hogy az arc alakját, arckifejezését és megjelenését w vektorként reprezentálják egy alacsony dimenziós latens térben $\mathbb{R}^d$ [9]. A vetítés (illetve rekonstrukció) ebből a latens térből lineáris vagy multi lineáris műveletekkel definiálható, és úgy is felfogható, mint a nagy-dimenziós információ kódolása (ill. dekódolása) a $\mathbb{R}^d$ térben.

A mély tanulás új eszközt nyújt a 3DMM-ek építéséhez, amely nemlinearitást használ mind a a kódolóban és a dekódolóban. Az ilyen *morphable* modellek létrehozása egy jelenleg nagyon aktív kutatási terület. Az alak és textúra modellezésre gyakran használt lineáris modellt felhasználva láthatjuk a *deep learning* és a hagyományos módszerekkel tanult kódoló és dekódoló közötti kapcsolatot.

A mély tanulás kontextusában egy ilyen lineáris modell az alábbi egyenletben formalizálva pontosan megfelel egy teljesen összekapcsolt rétegnek egy neurális hálózatban [9].

$$c(w) = \bar{c} + Ew \tag{7}$$

, ahol $\bar{c}$ a képzési adatokra számított átlag, $E \in \mathbb{R}^{3n*d}$ egy olyan mátrix, amely tartalmazza a d legdominánsabb sajátvektorjait
a formakülönbségekre $c_i - \bar{c_i}$ számított kovarianciamátrixban és w az alacsony dimenziós alakparaméter-vektor.

[9]Lényegében, a w paramétervektora bemeneti jellemzők szerepét játssza, az e_j főkomponensek pedig a súlyokét és az átlag $\bar{c}$ a torzítás. Ez úgy is felfogható, mint a dekódolás a latens paramétertérből a c adattérbe. Vetítés a modellre hasonlóképpen tekinthető egy teljesen összekapcsolt réteggel

történő kódolásnak, ahol a bemeneti jellemzők az adatok, a súlyok pedig a transzponált főkomponens mátrix sorai, az eltérések pedig adottak $-e_j^T \bar{c}$ által.

Az analógia lezárásaként a PCA (Principal Component Analysis) a kódoló és a dekódoló egyetlen rejtett réteggel rendelkező lineáris autokódolóvá (*autoencoder*) történő kombinálásával, végezhető el. Egy ilyen autokódoló d neuronokkal a rejtett rétegben egy olyan látens teret fog megtanulni, amelynek kiterjedése megegyezik a d dimenziós PCA-val, bár az ortogonalitás garanciája nélkül (ez megfelelő veszteségfüggvényekkel biztosítható).

5.2. Deep Face rekonstrukció

A következőkben a mély neurális hálózatokon alapuló sűrű monokuláris arcrekonstrukció megközelítéseket tárgyaljuk. Megbeszéljük a felhasznált képzési adatokkal szemben támasztott követelményeket, valamint a különböző képzési stratégiákat.

Nézzük meg először közelebbről a rekonstrukciós problémát. Blanz és Vetter [8] egy optimalizációs megközelítésen alapuló parametrikus modell illesztésével, azaz a gradiens süllyedéssel, kezeli a monokuláris arc rekonstrukcióját.

A mélytanulási megközelítések hasonló optimalizálási stratégiát követnek, de az optimalizálási probléma "tesztelés" idején történő megoldása helyett, például egy paraméterregresszort képeznek ki egy nagyméretű képadathalmaz alapján [9]. A regresszor úgy értelmezhető, mint egy kódoló hálózat, amely egy 2D-s képet bemenetként fogad, és az alacsony dimenziós arcreprezentációt adja ki.

A kódolók kombinálhatók klasszikus arcmodelleken alapuló dekódolókkal, hogy végponttól-végpontig tartó kódoló-dekódoló architektúrákat hozzanak létre. Ez a módszertan széles körben elterjedt, és lehetővé teszi a klasszikus modellalapú és mélytanulási megközelítések ötvözését.

5.2.1. Felügyelt rekonstrukció

Felügyelt regressziós megközelítések párosított tanítási adatok segítségével, azaz egy monokuláris képgyűjtemény és a megfelelő 3DMM paraméterei segítségével tanulnak [9].

Az egyik alapvető kérdés itt az, hogy hogyan lehet hatékonyan megszerezni az adatot egy ilyen felügyelt tanulási feladathoz. A következőkben

kategorizáljuk a megközelítéseket a képzési adatok alapján.

Az egyik lehetőség az lenne, hogy a felhasználók határozzák meg az adatot. Míg ez egy népszerű stratégia, amelyet gyakran alkalmaznak rekonstrukciós problémáknál [20], a sűrű geometria, a megjelenés és a helyszín megvilágítás pontos meghatározása szinte megoldhatatlan.

Hasonló megközelítést alkalmaztak például Olszewski [3] munkájában,ahol három professzionális animátor kézzel készítette el a *blendshape* animációt egy videokliphez illesztve. A sűrű rekonstrukciós feladatokhoz egyes megközelítéseket ellenőrzött, több nézetből készült felvételek alapján tanítanak be.

Így megkapható az adat egy több nézetből történő rekonstrukciós megközelítéssel, amelyet egy 3DMM illesztése követ. Ezáltal megkapjuk a 3D adatot. Általában az alapadatok nagyon jó minőségűek, de a monokulárisan rögzített képek eloszlása nem egyezik meg az *in-the-wild* adatokkal, ami általánosítási problémákhoz vezethet a tesztelés idejében.

Anh Tuan Tran megközelítése monokuláris rekonstrukciót végez ugyanarról a személyről készült több képre, és kiszámít egy konszolidált arcazonosságot a 3DMM paraméterek egyszerű átlagolása alapján.

Jelenleg a kutatóközösségben számos megközelítés szintetikus képzési adatokat használ tanításhoz, mivel könnyen beszerezhető és tökéletes annotációkkal rendelkeznek.

Adott egy 3DMM arc, véletlenszerű identifikációk és kifejezések mintavételezhetők a paramétertérben. Majd, a modelleket véletlenszerű megvilágítási körülmények között és különböző nézőpontokból lehet renderelni a monokuláris képek létrehozásához.

A háttértámogatást gyakran alkalmazzák úgy, hogy a generált arcokat a valós világ sokféle hátterére renderelik. Mivel az összes paramétert ellenőrzik, ezért azok kifejezetten ismertek, és alapigazságként használhatók.

Míg könnyen hozzájuthatunk a szintetikus képzési adatokhoz, gyakran van egy nagy tartományi rés a szintetikus és a valós világ képei között, ami súlyosan befolyásolja a a valós képekre való általánosítást. Például a hajat, az arcszőrzetet, a felsőtestet, vagy a száj belsejét gyakran egyáltalán nem modellezik. Az egyik lehetőség, a jövőben olyan modelleket használni, amelyek tartalmazzák ezeket, hogy ellensúlyozzuk ezt a problémát.

A valós és a szintetikus képzési adatok előnyeinek kihasználása érdekében

számos jelenlegi megközelítést e két terület adatainak keverékével tanítanak. A remény itt az, hogy a megközelítés megtanulja kezelni a valós világi képeket, miközben a szintetikus tréning adatok tökéletes alapigazsága segítségével stabilizálható a tanulás.

Ennek egy érdekes változata a tanítás önfelügyelt *bootstrappelése*. A további megközelítéseket, amelyek tanítása nem igényel alapigazságadatokat a következő fejezetben vizsgáljuk meg.

5.2.2. Önfelügyelt rekonstrukció

A konvolúciós neurális hálózatok felügyelt tanítása annotált adathalmazt igényel. A legtöbb eddig tárgyalt módszer ilyen - szintetikus vagy valós - adathalmazokat használ.

[9] A közelmúltban egyes megközelítések önfelügyeletet alkalmazó tanulást használtak, azaz 3D címkék nélküli, valós képi adathalmazokon történő tanítást. Ezt az analízis-szintézis és a mélytanulási technikák kombinációja tette lehetővé.

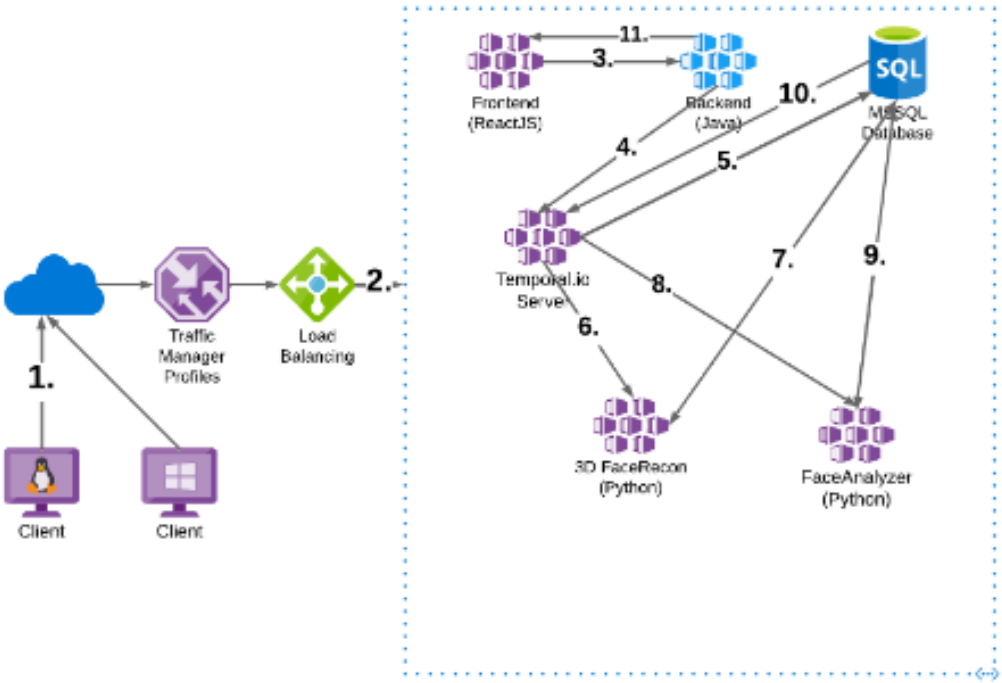
Tewari *et al.* [1] egy modell-alapú kódoló-dekódoló architektúrát mutatott be, amely a a betanítható dekódert egy fix dekóderrel helyettesíti. Ez a dekóder a 3DMM paramétereket (látens kód) mint bemenet, amelyet a kódoló jelez, a 3DMM segítségével 3D-rekonstrukcióvá alakítja. Továbbá szintetikus képet készít a rekonstrukcióról egy differenciálható renderelő segítségével. A rendereléshez szükséges külső paramétereket szintén a kódoló jelzi előre.

Az alkalmazott veszteségfüggvény nagyon hasonlít az analízis-szintézisben használthoz [9], ami fotometriai igazítást és statisztikai szabályozást foglal magába. Egy ilyen technikára úgy is gondolhatunk, mint egy közös *analysis-by-synthesis* optimalizálási probléma egy nagyméretű tanulási adathalmazon, egyetlen kép helyett. Ez lehetővé teszi a paraméterregresszor képzését 3D felügyelet nélkül.

6. Megvalósítás

Ebben a fejezetben a projekt¨unk redszerterv´et, g´epi tanul´asi pipeline-j´at, valamint kulcsfontoss´ag´u met´odusait mutatjuk be r´eszletesen.

6.1. Felhő alapú architektúra



14. Ábra. A rendszer fel´ep´ıt´ese

Ahogy kor´abban is eml´ıtett¨uk, a kit¨uz″ott c´eljaink k¨oz¨ott van, hogy a 3D arc rekonstrukci´o´es az arc anal´ızis el´erhet″o legyen felh″on kereszt ¨ul. Tov´abbiakban megvizsg´aljuk a rendszer fel´ep´ıt´es´et illetve m¨uk″od ´es´et a fenti´abra seg´ıts´eg´evel. Az´abr´an l´athat´o, hogy 6 k¨ul¨onb¨oz″o szolg´altat´ast M inikube lok´alis klaszteren¨uzemeltet¨unk.

A minikube egy helyi Kubernetes, amellyel gyorsan tudunk l´etrehozni egy lok´alis klasztert, amely megk¨onny´ıti a Kuberneteshez val´o tanul´ast´es fejleszt´est

A 6 k¨ul¨onb¨oz˝o szolg´altat´as az al´abbiak:

- Frontend
- Backend
- Temporal.io szerver
- 3D arcrekonstrukci´os modu
- Arc anal´ızis modul
- Adatb´azis

A T emporal.io egy olyan technol´ogia amely seg´ıts´eg´evel gyorsan ´es egyszer˝uen tudunk workf low-t implement´alni. Ehhez sz¨uks´eges a szerver, mert ott bonyol´odik le egy workf low sikeres elv´egz´ese, vagy hiba eset´en biztos´ıt egy dashboard-t amivel megtudjuk vizsg´alni a hiba ok´at. Illetve,a workf low-ban implement´altuk a SAGA programoz´asi mint´at, amely kezeli hib´ak eset´en az adatb´azissal kapcsolatos m˝uveleteket. Ezzel is robosztusabb´a teszi a rendszert.

Ha megn´ezz¨uk a fentí´abr´at, l´athatjuk hogy, a workf low bel´ep´esi pontja a HTTP k´er´es, aminek payload-ja, a k´ep egy c´elszem´elyr˝ol,´es amint a f rontend-t˝ol megkapja a k´er´est a backend, az elind´ıtja a workf low-t. A k¨ovetkez˝o l´ep´es hogy, a workf low met´odusai egym´as ut´an megh´iv´odnak. V´eg¨ul a felhaszn´al´o visszakapja a rekonstru´alt k´epet, illetve a k´epen l´ev˝o szem´ely akkorí´erzelmí´allapot´at´es´eletkor´at. Az arcrekonstrukcí´os´es arc anal´ızis moduljainkat P ython nyelven implement´altuk, mivel mesters´eges intelligenci´aval dolgozunk, a k¨ul¨onb¨oz˝o nagymennyis´eg˝u P ython k¨onyvt´ar el´erhet˝os´ege nagy m´ert´ekben megseg´ıtette munk´ankat.

6.2. Arcrekonstrukció

6.3. Programozási nyelv

A projektben használt programozási nyelvek:

6.3.1. Python

[23]A Python egy könnyen megtanulható, nagy teljesítményű programozási nyelv. Hatékony, magas szintű adatszerkezetekkel és egy egyszerű, de hatékony megközelítése az objektumorientált programozásnak. A Python elegáns szintaxisa és a dinamikus típusmeghatározás, valamint interpretált természetével együtt ideális nyelvvé teszi a szkriptek írásához és a gyors alkalmazásfejlesztéshez számos területen a legtöbb platformon.

A Python gépi tanulás specifikus könyvtárak és keretrendszerek széles választéka leegyszerűsíti a fejlesztési folyamatot és csökkenti a fejlesztési időt.

6.3.2. CUDA C++

2006 novemberében az NVIDIA® bemutatta a CUDA®-t, egy általános célú párhuzamos számítási platformot és programozási modellt, amely az NVIDIA GPU-k párhuzamos számítási motorját kihasználva számos összetett számítási feladatot hatékonyabban old meg, mint egy CPU-n a [22] szerint. A GPU a nagymértékben párhuzamos számításokhoz van specializálva, ezért úgy tervezték, hogy több tranzisztort fordítanak az adatfeldolgozásra, mint az adatok gyorsítótárazására és az adatáramlás-szabályozásra.

A projektünkben a nagyobb párhuzamositható komplex matematikai számitásokat a CUDA végzi, a fenti felsorolt erősségei miatt.

6.3.3. C++

A C++ egy objektumorientált programozási nyelv, amelyet Bjorne Stroustrop informatikus fejlesztett a C nyelvcsalád továbbfejlesztésének részeként, hogy a fejlesztőknek nagyobb irányításuk legyen a memória es a rendszer erőforrásai felett.

Általánosságban elmondható, hogy a C++ használata más nyelvek helyett a teljesítmény miatt indokolt. Ennek oka az, hogy a C++ olyan absztrakciós lehetőségeket kínál, amelyeknek nincs teljesítmény-többletterhelése futásidőben.

6.4. Tanítási adatok

Minden tanulási algoritmus alapvető része a betanító adathalmaz. Megfelelő méretű, minőségi tanító adatbázis nélkül a tanulási modell nem tudná karakterizálni az adatokat. A CNN-ek esetében, amelyeknél általában több millió paramétert használnak, az általánosítási probléma még összetettebb, és gyakran egy modell betanításához nagymértékű adathalmazra van szükség.

Azonban a meglévő 3D arcokat tartalmazó adathalmazok általában csak néhány száz alanyból állnak, ami alkalmatlanná teszi őket a mélytanulási (deep learning) feladatokra. Ideális esetben elég 3D-s arcot rögzíthetnénk és ezek segítségével betaníthatnánk a CNN modellt, viszont az arcok millióinak pontos mélységérzékelővel történő szkennelése jelenleg nem kivitelezhető.

Alternatív megoldásként vehetünk egy sor 2D-s képet, és egy rekonstrukciós algoritmust alkalmazva generálható a geometriai reprezentáció. Mindazonáltal egy ilyen megközelítés a rekonstrukciós lehetőségeinket az általunk használt konkrét geometria-rekonstrukciós algoritmusra korlátozná.

Ehelyett *Elad et al.* [25] azt javasolja, hogy közvetlenül generáljunk különböző geometriákat egy *morphable* modell segítségével. Minden ilyen geometriát ezután véletlenszerű megvilágítási körülmények között renderelünk, és a képsíkra vetítünk. Így kapunk egy adag képet amelyekhez ismert a valódi geometriájuk.

6.5. Gépi tanulási keretrendszerek

Ebben a fejezetben a deep learning keretrendszereket vizsgáljuk meg és a továbbiakban belemegyünk egyes keretrendszerek részleteibe, hogy megnézzük melyik keretrendszer felel meg az igényeinknek.

6.5.1. Keras

[26]A Keras egy kompakt és könnyen tanulható, magas szintű Python könyvtár a mélytanuláshoz, amely a TensorFlow (vagy a Theano vagy a CNTK) keretrendszerek mellet is futhat.

Lehetővé teszi a fejlesztők számára, hogy a mélytanulás fő fogalmaira, például a neurális hálózatok rétegeinek létrehozására összpontosítsanak, miközben a tenzorok, alakjaik és matematikai részleteik apró részleteiről a Keras gondoskodik. A TensorFlow (vagy Theano vagy CNTK) kell, hogy legyen a Keras back endje. A Keras-t mélytanulási alkalmazásokhoz a viszonylag összetett TensorFlow-val (vagy Theanóval vagy CNTK-val) való interakció nélkül is használhatja.

6.5.2. PyTorch

A PyTorch egy optimalizált tenzorkönyvtár, amelyet elsősorban GPU-kat és CPU-kat használó Deep Learning alkalmazásokhoz használnak. Egy nyílt forráskódú gépi tanulási könyvtár Pythonhoz, amelyet főként a Facebook AI Research csapata fejlesztett ki. A PyTorch a python és a torch könyvtárra épül, amely támogatja a tenzorok számítását GPU-n.

6.5.3. TensorFlow

A Google Brain csapata által létrehozott TensorFlow egy nyílt forráskódú könyvtár numerikus számításokhoz és nagyméretű gépi tanuláshoz. A TensorFlow a gépi tanulás és a mélytanulás (más néven neurális hálózat) modelljeinek és algoritmusainak egész sorát foglalja össze. A Python nyelv segítségével egy kényelmes front-end API-t biztosít a keretrendszerrel történő alkalmazásépítéshez, miközben ezeket az alkalmazásokat nagy teljesítményű C++ nyelven hajtja végre.

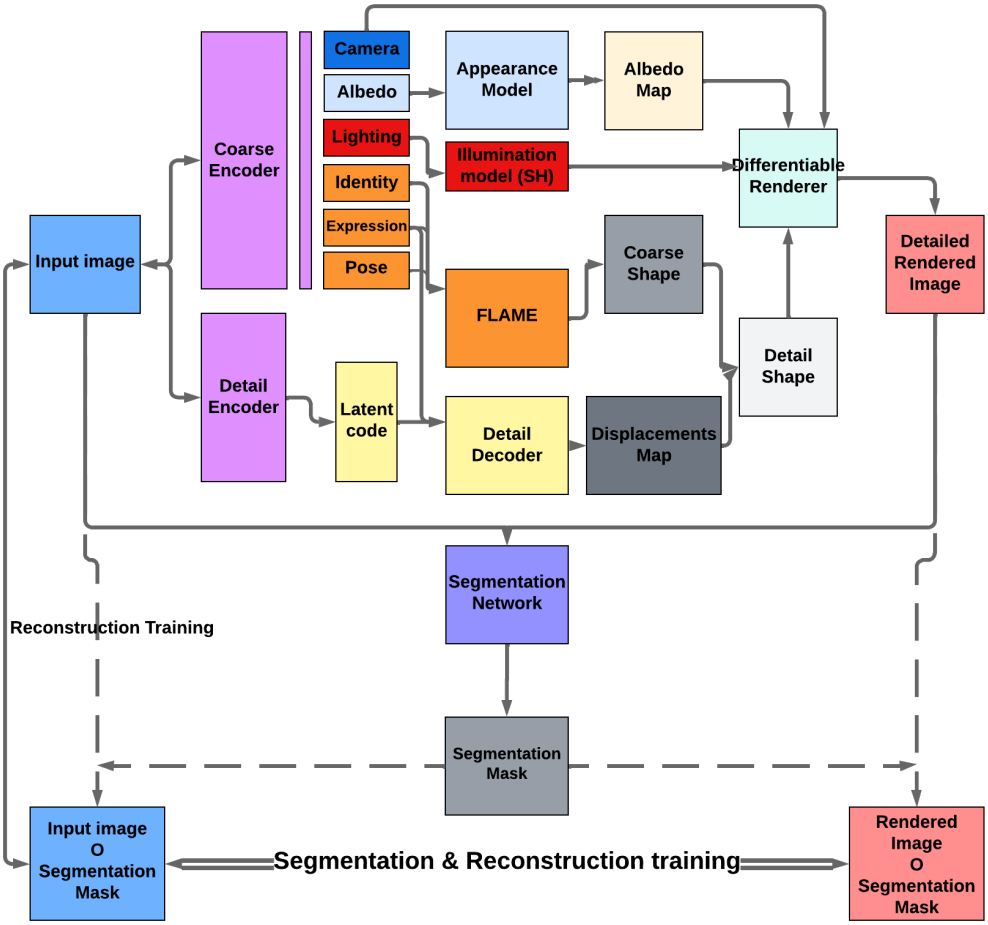
6.5.4. Konkluzió

Az alábbi táblázatban[1] összefoglaljuk a felsorolt keretrendszerek főbb tulajdonságait.

Keretrendszerek			
	Keras	**PyTorch**	**TensorFLow**
API szint	Magas	Alacsony	Magas és Alacsony
Architektúra	Egyszerű, tömör, olvasható	Összetett, kevésbé olvasható	Nem könnyű használni
Adatkészletek	Kisebb adatkészletek	Nagy adathalmazok, nagy teljesítmény	Nagy adathalmazok, nagy teljesítmény
Debug	Egyszerű hálózat, így a debug nem gyakran szükséges	Jó debugging képességek	Nehéz debugging
Népszerűség	Legnépszerűbb	Harmadik legnépszerűbb	A második legnépszerűbb
Sebesség	Lassú, alacsony teljesítmény	Gyors, nagy teljesítményű	Gyors, nagy teljesítményű

A keretrendszereket megvizsgálva arra a döntésre jutottunk, hogy a projektben a PyTorch keretrendszerét fogjuk implementálni, mivel a PyTorch könnyebben tanulható és könnyebb vele dolgozni a többihez viszonyvita, valamint gyors es nagy teljesítményt biztosít, illetve egy egyszerűen használható API-t nyújt a CPU-n generált tenzor GPU-ra történő átviteléhez.

[1] https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article



15. Ábra. A h´al´ozat tan´ıt´asi pipeline-ja.

Hivatkozások

[1] Tewari, A., Zollhofer, M., Kim, H., Garrido, P., Bernard, F., Perez, P., Theobalt, C. (2017). *Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 1274-1283)*

[2] Tuan Tran, A., Hassner, T., Masi, I., Medioni, G. (2017). *Regressing robust and discriminative 3D morphable models with a very deep neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5163-5172)*

[3] Olszewski, K., Lim, J. J., Saito, S., Li, H. (2016). *High-fidelity facial and speech animation for VR HMDs. ACM Transactions on Graphics (TOG), 35(6), 1-14*

[4] Styner, M. A., Rajamani, K. T., Nolte, L. P., Zsemlye, G., Székely, G., Taylor, C. J., Davies, R. H. (2003, July). *Evaluation of 3D correspondence methods for model building. In Biennial International Conference on In- formation Processing in Medical Imaging (pp. 63-75). Springer, Berlin, Heidelberg*

[5] Morales, A., Piella, G., Sukno, F. M. (2021). *Survey on 3D face reconstruction from uncalibrated images. Computer Science Review, 40, 100400.*

[6] Kala, R. (2016). *On-road intelligent vehicles: Motion planning for intelligent transportation systems. Butterworth-Heinemann.*

[7] Kovacs, L., Zimmermann, A., Brockmann, G., Gühring, M., Baurecht,H., Papadopulos, N. A., ... Zeilhofer, H. F. (2006). *Three-dimensional recording of the human face with a 3D laser scanner. Journal of plastic, reconstructive and aesthetic surgery, 59(11), 1193-1202*

[8] Blanz, V., Vetter, T. (1999, July). *A morphable model for the synthesis of 3D faces. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques (pp. 187-194)*

[9] Egger, B., Smith, W. A., Tewari, A., Wuhrer, S., Zollhoefer, M., Beeler, T.,... Vetter, T. (2020). *3d morphable face models—past, present, and future. ACM Transactions on Graphics (TOG), 39(5), 1-38*

[10] Cashman, T. J., Fitzgibbon, A. W. (2012). *What shape are dolphins? building 3d morphable models from 2d images. IEEE transactions on pattern analysis and machine intelligence, 35(1), 232-244*

[11] Ackermann, J., Goesele, M. (2015). *A survey of photometric stereo techniques. Foundations and Trends® in Computer Graphics and Vision, 9(3-4), 149-254, 149–254.*

[12] Hernández, C., Vogiatzis, G., Brostow, G. J., Stenger, B., Cipolla, R. (2007, October). *Non-rigid photometric stereo with colored lights. In 2007 IEEE 11th International Conference on Computer Vision (pp. 1-8). IEEE*

[13] Nehab, D., Rusinkiewicz, S., Davis, J., Ramamoorthi, R. (2005). *Efficiently combining positions and normals for precise 3D geometry. ACM transactions on graphics (TOG), 24(3), 536-543.*

[14] Patel, A., Smith, W. A. (2012). *Driving 3D morphable models using shading cues. Pattern Recognition, 45(5), 1993-2004*

[15] Abraham, A. (2005). *Artificial neural networks. Handbook of measuring system design*

[16] Jain, A. K., Mao, J., Mohiuddin, K. M. (1996). *Artificial neural networks: A tutorial. Computer, 29(3), 31-44*

[17] *MARCELL, Borza. Mesterséges neurális hálózatok matematikai alapjai.*

[18] Tamás, K. (2002). *A mesterséges neurális hálók a jövőkutatás szolgálatában.*

[19] Krenker, A., Bester, J., Kos, A. (2011). *Introduction to the artificial neural networks. Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech, 1-18.*

[20] Saragih, J. M., Lucey, S., Cohn, J. F. (2011, March). *Real-time avatar animation from a single image. In 2011 IEEE International Conference on Automatic Face and Gesture Recognition (FG) (pp. 117-124). IEEE*

[21] Indolia, S., Goswami, A. K., Mishra, S. P., Asopa, P. (2018). *Conceptual understanding of convolutional neural network-a deep learning approach. Procedia computer science, 132, 679-688.*

[22] NVIDIA *CUDA C++ Programming Guide, PG-02829-001 v11.5 November 2021*

[23] Van Rossum, G., Drake Jr, F. L. (1995). *Python reference manual. Amsterdam: Centrum voor Wiskunde en Informatica.*

[24] de Souza, R. L., Maciel, C., dos Santos Nunes, E. P. (2021, November). *Inspeção semiótica no sistema do Metahuman Creator: avatares em foco. In: Anais da XXI Escola Regional de Informática de Mato Grosso. SBC, 2021. p. 77-83.*

[25] Richardson, E., Sela, M., Kimmel, R. (2016, October). *3D face reconstruction by learning from synthetic data. In 2016 fourth international conference on 3D vision (3DV) (pp. 460-469). IEEE.*

[26] Manaswi, N. K. (2018). *Understanding and working with Keras. In Deep Learning with Applications Using Python (pp. 31-43). Apress, Berkeley, CA.*

[27] Guo, Y., Cai, J., Jiang, B., Zheng, J. (2018). *Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. IEEE transactions on pattern analysis and machine intelligence, 41(6), 1294-1307.*

[28] Richardson, E., Sela, M., Kimmel, R. (2016, October). *3D face reconstruction by learning from synthetic data. In 2016 fourth international conference on 3D vision (3DV) (pp. 460-469). IEEE.*

[29] Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., Tong, X. (2019). *Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0)*

[30] Chen, D., Hua, G., Wen, F., Sun, J. (2016, October). *Supervised transformer network for efficient face detection. In European Conference on Computer Vision (pp. 122-138). Springer, Cham.*

[31] Kingma, D. P., Ba, J. (2014). *Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.*

[32] Feng, Y., Feng, H., Black, M. J., Bolkart, T. (2021). *Learning an animatable detailed 3D face model from in-the-wild images. ACM Transactions on Graphics (TOG), 40(4), 1-13.*

[33] Li, C., Morel-Forster, A., Vetter, T., Egger, B., Kortylewski, A. (2021). *To fit or not to fit: Model-based Face Reconstruction and Occlusion Segmentation from Weak Supervision. arXiv preprint arXiv:2106.09614.*

[34] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero (2017). *Learning a model of facial shape and expression from 4D scans. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia) 36, 6 (2017), 194:1–194:17.*

[35] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. (2017). (2017). *Avatar Digitization from a Single Image for Real-time Rendering. ACM Transactions on Graphics (TOG) 36, 6 (2017), 195:1–195:14.*

[36] Adrian Bulat and Georgios Tzimiropoulos (2017). *How far are we from solving the 2D and 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks). In International Conference on Computer Vision.*

[37] Mohammed, S. B., Abdulazeez, A. M. (2021). *Deep Convolution Neural Network for Facial Expression Recognition. PalArch's Journal of Archaeology of Egypt/Egyptology, 18(4), 3578-3586.*

[38] Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). *Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4690-4699).*

[39] Yan, Y., Lu, K., Xue, J., Gao, P., and Lyu, J. (2019, July). *Feafa: A well-annotated dataset for facial expression analysis and 3d facial animation. In 2019 IEEE International Conference on Multimedia and Expo Workshops (ICMEW) (pp. 96-101). IEEE.*

[40] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. (2009, September). *A 3D face model for pose and illumination invariant*

face recognition. In 2009 sixth IEEE international conference on advanced video and signal based surveillance (pp. 296-301). Ieee.