**PROJEKTMUNKA 2**

**OE-NIK** Hallgató neve: **Szücs Klaudia**

**2021** Hallgató törzskönyvi száma: **T/006399/FI12904/N**

**Tartalomjegyzék**

1. **A megoldandó probléma felvezetése**
   1. Pajzsmirigy és betegségei.
2. **Irodalomkutatás**
   1. Gépi tanulás és mélytanuló algoritmusok bevezető
   2. Konvolúciós neurális hálózatok
   3. A Konvolúciós neurális hálózatok építő kövei
   4. Mélytanuló algoritmusok egészségügyi felhasználása képfeldolgozás terén
   5. Konvolúciós neurális hálózati architektúrák a probléma szempontjából
3. **Technológiák kiválasztása**

3.1 R-CNN

3.2 Fast R-CNN

3.3 Faster R-CNN

3.4 Mask R-CNN

1. **Irodalomjegyzék**
   1. **A megoldandó probléma felvezetése és célkitűzések**

A pajzsmirigy az egyik legfontosabb belső elválasztású mirigyünk. Működése elengedhetetlen a normális élet fenntartásához. Azonban még a 21. században is különböző funkció zavarainak felismerése, esetleges fizikai elváltozásának észlelése igen bonyodalmas, és kellő nagyságú szaktudást igénylő folyamat. Endokrinológiai szempontból a második legnagyobb problémát generálják világszerte szervezetünkben a pajzsmirigy működésének zavarai. Diagnózis felállításhoz elengedhetetlen CT, MRI képek készítése vagy ultrahangos vizsgálat.

Mivel főként orvos tapasztalatán múlik korai észlelése a betegségnek és pontossága, ezért egy olyan webes alkalmazást szeretnék létrehozni, ami képes felismerni a különböző elváltozásokat ultrahangos képek alapján, és besorolni 5 különböző kategoriába, normál, jóindulatú, félreérthető/határozatlan, gyanús, rosszindulatú. Az objektum detektálásra egy finoman hangolt konvolúciós neurális hálót fogok használni.

**1.1 A pajzsmirigy**

A pajzsmirigy a nyak elülső részén a gége, valamint a légcső felső része előtt elhelyezkedő kétlebenyes belső elválasztású mirigy.

A picture containing text

Description automatically generated

**1.ábra Pajzsmirigy felépítése**

A pajzsmirigy működését közvetlenül az agyalapi mirigy elülső részében termelődő TSH -hormon segítségével szabályozza. Elsődleges feladata a szervezet anyagcseréjének szabályozása, illetve a növekedést befolyásoló hormonok termelése. Azonban szinte mindegyik szervünkre valamilyen hatással van, ezért megfelelő működése kiemelten fontos. Tiroxin és trijód-tironin, jód tartalmú hormonok termelődnek főként e mirigyünkben, amiknek fontos szerepe van egészséges metabolizmus fenntartásában. Emellett hatással van a fehérje termelődésre, egyezséges testhőmérséklet fenntartására, energiatermelésre, azaz ahogy fentebb említettem az anyagcserét gyorsítják, illetve szívverés növelésével, légzés növelésével fokozzák a mitokondriumok aktivitását. Pajzsmirigy bármilyen csekély méretű elváltozása is hatással van egész szervezetünkre. Ezért fontos kezdeti szakaszban való felismerésük, mivel könnyen elhatalmasodhat a probléma. Betegségek nehezen észrevehetőek, és számos vizsgálatot igényel felismerésük. Funkcionális zavarai közé tartozik például a túlműködés, ami TSH-hormon alul termelődése vált ki, vagy ellenkező esetben, amikor túl sok termelődik pajzsmirigy alul működés jellemző. Mindkettő betegség népbetegség nevezhető manapság már. Túl működés esetén kialakulhat úgynevezett Basedow Graves-kór, ami egy autoimmun betegség, illetve alul működés esetén jellemző betegség a Hashimoto-kór. Mindkettő esetben a pajzsmirigy fizikailag megváltozik, így megfelelő vizsgálatokkal, mint például CT észlelhető a probléma. Ezenkívül szóba jöhetnek koros megnagyobbodások, azaz golyvák, göbök, vagy másnéven nodulák és a tumor, amik szintén jól diagnosztizálhatók CT, vagy MRI segítségével.

**2. Irodalomkutatás**

**2.1 Gépi tanulás és mélytanuló algoritmusok bevezető**

A gépi tanulás az elmúlt évszázadban drámai fejlődésen ment keresztül, ami az élet minden területére kihatással volt. Technológiája a modernkori társadalom számos területén gondolkodás módját befolyásolni képes a közösségi hálókon megjelenő tartalomszűréseken át a webes kereséseseket, és nem beszélve az online kereskedelemről. Határozottan egyre inkább meghatározó szerepet tölt be mindennapi élet területén. Mind ezen csekély manapság már hétköznapinak nevezhető dolgok hátterében gépi tanuló rendszerek állnak. A mesterséges neurális hálózatók, másnéven mélytanuló algoritmusok lehetővé teszik a számítógép számára, hogy összetett, bonyolult adathalmazokban ismétlődések fedezzenek fel. Ezeket a tanuló rendszerek főként képeken való objektum detektálásra, beszéd szöveggé alakítására, híreket, posztokat vagy termékeket a felhasználó érdeklődési körére szabására használják. Azonban a kezdetekben, amikor még nem voltak megfelelő erőforrások, a hagyományos gépi tanulási technikák csak korlátozott számban voltak képesek természetes adatok nyers formában történő feldolgozására. Egy darab mintafelismerő rendszer megépítéséhez mérnöki munkára és kellően nagy szakismeretre volt szükség. Azonban az elmúlt évek során rengeteg kutatás folyt a gépi tanulási technikák tervezéséről, és a tanulmány konklúziója, hogy jobb minta felismerő rendszereket lehet tervezni, ha jobban támaszkodunk az automatikus tanulásra és kevésbé a kézzel tervezett heurisztikákra. Ezt a gépi tanulás modellek időarányos fejlődése, illetve a mélytanuláshoz szükséges számítógépes infrastruktúra (hardveres és szoftveres) rohamos mértékű fejlődése tette lehetővé.

Mélytanuló algoritmusokat mesterséges neurális hálókból fejlesztették ki, és mostanra legelterjedtebbek a gépi tanulás területén. Mesterséges neurális hálókkal kapcsolatos kutatások már az 1940-es években elkezdődtek. A mélytanulási modellek általában hierarchikus struktúrát alkalmaznak a rétegek összekapcsolására. Az alsó rétegek kimenete lehet egy magasabb réteg bemenete, lineáris vagy nemlineáris számítások segítségével átalakított verziója. Tehát a számítógépek a hasznos reprezentációkat és jellemzőket automatikusan, közvetlenül a nyers adatokból tanulják meg, megkerülve a kézi feldolgozású nehézkes lépést. Ezek a modellek képesek már az alacsony színtű jellemzőket magasabb szintű absztrakt jellemzőkké átalakítani. Ennek a tulajdonságnak köszönhetően mélytanulási modellek erősebbek lettek, mint a gép tanulási modellek a jellemzők reprezentálásában. Az orvosi képalkotásban a mélytanulást iránti érdeklődést leginkább a konvolúciós neurális hálózatok keltették fel. Ezekkel az algoritmusokkal lehetséges képek és más strukturált adatok hasznos reprezentációit megtanulni. Amint lehetővé vált a közvetlenül adatokból való jellemzők kinyerése, és tanításra való felhasználása, más technikákat a háttérbe szorítottak.

**2.2 Konvolúciós Neurális Hálózatok (Convolutional Neural Networks)**

A konvolúciós hálózatok, vagy másnéven konvolúciós neurális hálózatok az adatok feldolgozására szolgáló neurális hálózatok egy speciális fajtája, amely ismert rácsszerű topológiával rendelkezik. Ezeket a hálózatokat úgy tervezték, hogy automatikusan és adaptívan megtanulják a jellemzők térbeli hierarchiáját a backpropagation algoritmuson keresztül, több építőelem, például konvolúciós rétegek, pooling rétegek és teljesen összekapcsolt rétegek felhasználásával. Az első kettő, a konvolúciós rétegek és a pooling rétegek a jellemzők kinyerését végzik, míg a harmadik réteg kinyert jellemzőket a végső kimenetre, például a csoportosításra képezi le. A konvolúciós rétegek kulcsszerepet játszanak e féle hálózatok kialakításában, mivel ezek a rétegek matematikai folyamatok halmazából áll, a lineáris műveletek egy speciális típusából. A digitális képek pixelértékeit egy kétdimenziós rácsban, azaz számmátrixban tárolódnak, és egy kernelnek (kernel funkciónak) nevezett paraméterrácsot alkalmaznak minden egyes pixelen, ami rendkívül hatékonnyá ezt a technikát, mivel egy jellemző bárhol előfordulhat a képen. Ahogy az egyik réteg a másikba táplálja a csoportosított jellemzőket, hierarchikusan és fokozatosan egyre komplexebbé vállnak. A paraméterek, mint például a kernelek optimalizálásnak folyamatát nevezzük képzésnek, amely a csomópontok súlyaihoz kiszámítja a célfüggvény gradiens süllyedések, egy optimalizáló algoritmussal (backpropagation), amelynek feladata a kimenete minimalizálása, azaz a súlyok javítása.

**2.3 A Konvolúciós Neurális Hálózat építő kövei**

A konvolúciós neurális hálók felépítése előre csatolt neurális hálózatokéval hasonló.

Ahogy fentebb is említettem, a tulajdonságok kinyerését végző réteget, konvolúciós rétegnek nevezzük, ami jellemzően lineáris, illetve nemlineáris műveletek, azaz konvolúciós folyamat és az aktivációs függvény kombinációjából áll. Maga a konvolúció a lineáris műveletek, egy olyan speciális típusa, amelyet a főbb tulajdonságok kinyerésére használnak. A bemenet egy számokból álló mátrix, a tenzor, amiből a bemenet egy adott részén csoportosítva a főbb jellemzőket egy kisebb számmátrixot képez le, ezt nevezzük kernelnek. A kernel minden egyes elemén és az input tenzor közötti elemenként vett szorzatot, a tenzor minden egyes helyére ki számítjuk, majd ha ezeket összegezzük kapunk egy úgynevezett jellemzőtérképet (feature map), ami igazából megfelel az output tenzor értékének.

A diagram of a house

Description automatically generated with low confidence

**2.ábra**

Ezt a folyamatot több kernel alkalmazásával megismételjük, hogy tetszőleges számú jellemzőtérképet képezzünk le, amelyek a bemeneti tenzorok különböző jellemzőit reprezentálják. A konvolúciós műveleteknek két meghatározó paramétere van, amik a tenzorba kerülnek, a szűrők száma(j), és a rétegek száma(i). W (j, i). A bementi mátrix méretei különböző nagyságúak lehetnek. Azonban a konvolúciós művelet nem engedi, hogy egyes kernelek középpontja átfedje az input tenzorok legkülső elemét, ezzel csökkentve az output jellemzőtérképet. Ennek a problémának a kiküszöbölésére szolgál a null feltöltés, amivel a mátrix képes megtartani a síkbeli dimenziók számát. Ha ezt a lépést kihagynánk, egymásra épülő jellemző térképeket fokozatosan kisebbek lennek a rajtuk elvégzett matematikai műveletek után.

Diagram

Description automatically generated

**3.ábra Jellemző térkép leképzése**

A konvolúciós rétegből származó jellemzőtérképeket, ezután nemlineáros aktiválási függvényeken mennek keresztül, úgynevezett **aktivációs réteg**en. Ezáltal lehetővé válik, hogy a teljes neurális hálózat nemlineáris függvényekre való közelítése. Bár a közelmúltban a

biológiai neuronok viselkedésének egzakt matematikai reprezentációit használták, a szigmoid és a hiperbolikus érintő függvényt. Napjainkra a legelterjedtebbé lineáris aktiváló függvény a rectified linear unit, azaz ReLu, nőtte ki magát. Hiszen kiszámítása relatív egyszerűnek bizonyult a **ReLu (z) = max(0,z)** képlet segítségével.

Chart, histogram

Description automatically generated

**4.ábra Aktivációs függvények**

Ezután újabb jellemzőtérképek képződni, amiket egy **pooling réteg**ben összesítenek. Az integrálási műveletek kis rácsrégiókat, azaz rácselemeket vesznek bemenetként, és mindig egyes régióhoz egyetlen számot állítanak elő. A pooling műveletek legnépszerűbb formája a max pooling, amely a bemeneti jellemzőtérképből foltok kivonásával megkeresi a köztük lévő maximális értéket, és az összes többit elveti, úgyhogy dimenziók mélységet nem változtatja. A másik szóba jövő függvény az átlag pooling, ahol a magasságág szor szélesség méretű jellemzőtérképes egy 1x1 mátrixra kicsinyítjük le, oly módon, hogy az összes elem átlagát vesszük, miközben a dimenziók mélysége szintén nem változik.

Diagram

Description automatically generated

**5.ábra**

Ezt a műveletet jellemzően egyszer szokás használni a teljesen összefüggő réteg előtt. Ezen folyamatok célja a downsampling (lemintavételezés) művelet végrehajtása, amely következtében csökken a kimenő mátrix mérete, illetve a következő réteg input paramétereinek száma, úgyhogy a kis elmozdulások és torzulásokra transzlációs invarianciát vezetnek be. Majd az utolsó pooling réteg jellemzőtérképeit egy dimenziós vektorrá alakítják, ellaposítják, és egy vagy több teljesen összekapcsolt réteghez csatlakoztatják, amelyben minden bemenet minden kimenethez tanulható sulyokkal kapcsolódik. Konvolúciós rétegek által kivont, illetve pooling rétegek által lemintavételezett (downsampled) tulajdonságok létrejöttét követve a teljesen összekapcsolt rétegek egy részhalmaza képezi le őket a hálózat végső outputjaira.

Diagram

Description automatically generated

**6.ábra Konvolúciós neurális hálózati architektúra**

Új és tovább fejlesztett konvolúciós neurális hálózati architektúrák tervezése ezeket a komponenseket egyre bonyolultabb és összekapcsoltabb módon kell kombinálni. Amikor egy adott feladathoz szeretnék kiválasztani, vagy fejleszteni egy CNN architektúrát több tényezőt is figyelembe kell venni. Először is nagyon fontos alapkőnek számít magának a megoldandó feladatnak a precíz megértése, majd, hogy hogyan lehet a legjobban betáplálni a különböző adatok az architektúra hálózatába. Majd nem utolsó sorban a számítási és memóriafogyasztási probléma minél optimalizáltabb megoldására kell törekedni.

**2.3 Deep learning egészségügyi felhasználása képfeldolgozás felől**

Mélytanulási módszereket egyre gyakrabban használják klinikai gyakorlatok javítására. Alapvetően a képfeldolgozásnak több módja is van. A legismertebb feladatnak, a képosztályozást tekinthetjük, melynek célja egy bizonyos adathalmazban minden egyes képet

két vagy több kategória egyikébe besorolni. Ha például nem csak abban vagyunk érdekeltek, hogy mi van a képen, hanem abban is, hogy az hol helyezkedik el, akkor objektum lokalizációról beszélünk már. Azonban, ha ezt a kettőt ötvözzük, azaz ha objektum megtalálására és kategorizálására is szükségünk van akkor az objektum detektálás. Az orvosbiológiai képeken egy detektálási technikát is végeznek, hogy dobozkoordináták formájában azonosítsák azokat a területeket, ahol a beteg elváltozásai találhatók. Maga a mélytanulás alapú objektum detektálás 2 típusból állhat. Az egyik típus a régió-javaslat alapú lgoritmusok. Ez a megközelítés különböző típusú foltokat von ki a bemeneti képekből egy szelektív keresési algoritmus segítségével. Ezt követően a betanított modell eldönti, hogy az egyes foltokban több objektum létezik-e, és az objektumokat az érdeklődési régió (ROI) alapján osztályozza. A többi technika egylépcsős hálózatként, regressziós módszerrel végzi az objektumdetektálást. Ezek a megközelítések közvetlenül a teljes kép képpontjaiból keresik és detektálják a határoló doboz koordinátáit és az osztály valószínűségeit. A lentebb említett YOLOv3 is ilyen elven működik. Egy lépcsős hálózatoknak sokkal jobb a sebessége.

**2.4 Konvolúciós neurális hálózati architektúrák probléma szempontjából**

A modernebb mély tanulás kezdetleges időszakában elkezdtek architektúrákat ötvözni. Majd a későbbi hálózati architektúrák már folyamatosan kombináltabbá váltak, hiszen korábbi architektúrák felismeréseire és ötleteire épültek. Néhány mély tanulási módszert alkalmaztak a pajzsmirigy csomók ultrahangos képein, a finom hangolt GoogleNet modell segítségével, azonban a göbök automatikus detektálása nem sikerült. Egy másik tanulmány a tovább fejlesztett Fast R-CNN modellt ajánlotta. Garantált siker a pajzsmirigy göbök felismerésében, azonban egy optimalizáltább, több paraméteres megoldásra lenne szükség, így jött képbe a YOLOv3-DMRF nevű mélytanulási architektúra, ami a képes a pajzsmirigy csomók rendkívül gyors, illetve automatikus detektálására.A többi objektumfelismerő rendszerrel összehasonlítva a YOLO legkiemelkedőbb jellemzője a nagy hatékonyság.A YOLO („You Only Look Once”) modellek egy végponttól végpontig (end-to-end) tartó mélytanulási modellek. Félépítésükben több konvolúciós műveleteteket és pooling rétegeket tartalmaznak. Korábbi módszerek, mint például a fentebb említett R-CNN modellek is, több ezer hálózati kiértékelést igényelnek ahhoz, hogy egy képre való előrejelzést készítsenek, aminek optimalizálása is jóval bonyolultabb. Míg ezek a módszerek többlépcsős folyamatok voltak, a YOLO egyesíti a különböző folyamatokat, és egyetlen neurális hálózatot használ, ezáltál erősen optimalizált teljesítményre lehetünk figyelmesek, nem is említve a futási idő javulását. YOLOv3 amely már harmadik verziója, a nagy hatékonyság mellett már javított pontosságot is mutat. A modell a DarkNet-et veszi alapul, ami eredetileg 53 réteggel rendelkező neurális hálózatok képzésére szolgáló keretrendszer. A

detektálás elvégzéséhez újabb 53 réteget helyeztek rá, így összesen 106 rétegű teljesen konvolúciós architektúra jött létre. Előző verzióhoz hasonlítva azonban, a sebesége lecsökkent, de sokkal pontosabban képes apróbb objektumok precíz detektálására. Maga a detektálás egy 1x1 -es detektáló kernel segítségével történik, három különböző méretű jellemző térképen, a hálózat három különböző helyen. A detektáló kernel alapja:

**1 x 1 x (B + (5 + C))**,

ahol B, az előre megbecsülhető határoló dobozok száma, az 5-ös szám a négy határoló doboz attribútumaira, illetve az egy objektum bizalomra, vagy másnéven bizalmi pontszámra utal, azaz, hogy mennyire valószínű, hogy a tartalma objektumot, vagy nem. C pedig az osztályok száma. Az algoritmus három skálán készít előrejelzéseket, amelyeket az adott input kép dimenzióinak 32,16, és 8 mintavételezése add meg. A YOLO v3 összesen 9 horgonyzó dobozt (anchor box) használ. Minden skálázáshoz hármat. K-Means algoritmus segítségével megfelelően klaszterezéssel létrehozható a 9 horgonykocka. Az első skálához tartozik a három legnagyobb, a másodikhoz a három közepes, és az utolsóhoz a három legkisebb keret. Azaz dimenziók szerint csökkenő sorrendbe vannak rendezve. Így minden réteg a nagy, közepes vagy kis objektumok felismerésében jeleskedik.

**3.Technológiák kiválasztása**

Konvolúciós neurális hálózatok

* 1. **R-CNN**

A gépi látás terjedésével, egyre nagyobb hangsúly helyeződött az objektumdetektálás témakörére. Az első igazi áttörés a területen 2014-ben következett be, az RCNN publikálásával, melynek sokkal magasabb performancia szintje volt, az akkori más elterjedt objektumdetektáló metodikákkal szemben. Az addigi próbálkozások nagy komplexitású- és számításigényűek voltak, ami a detektáláshoz szükséges régiók kiszámításának költséges műveletének volt köszönhető. Bemeneti képeken külön elhelyezkedéssel és mérettel találhatóak objektumok, ebből következően a régiók meghatározásának folyamata különböző méretű, és nagyon nagy mennyiségű régió detektálását és kiértékelését igényli.

Erre a problémára ad tehát megoldás a RCNN (Regions with CNN). Az RCNN pipeline első lépése a „régiójavaslatok” létrehozása, azaz olyan régiók létréhozása, amelyek egy objektumhoz tartoznak. Egy Szelektív Kereső algoritmus a fő működési motorja, ami úgy működik, hogy a kép olyan alszegmentációt hozza létre, amelyek egy objektumhoz tartoznak – szín, textúra, méret és alak alapján –, és a hasonló régiókat iteratív módon kombinálja objektumok kialakításához. Ezen kereső algoritmus segítségével ~2000 darab független régiójavaslatot generál, amiken később egy konvolúciós neurális hálózat segítségével minden egyes régiójavaslathoz 4096 dimenziós jellemzővektort kapunk. Jellemzők kinyerése után, azoknak feldolgozása fog történni egy egyedi lineáris SVM (Support Vector Machine) nevű klasszifikációs módszerrel. Ez e a módszer a régiók közötti legjobb határvonal megtalálására ad megoldást, tehát probléma szempontjából besorolja egy adott osztályba, vagy jelzi annak a hiányát.

A close-up of a calculator

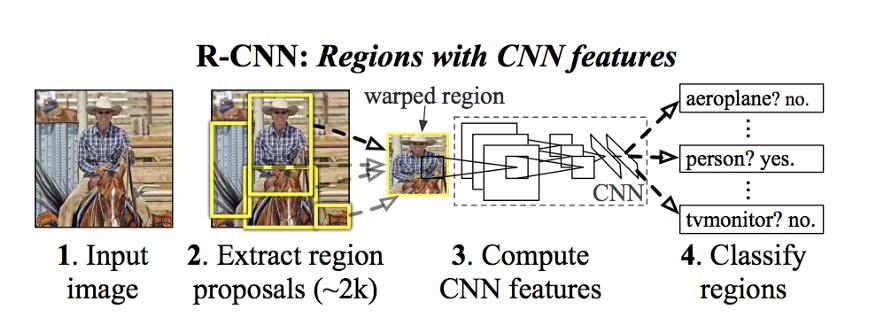
Description automatically generated with low confidence

**7.ábra SVM működése**

A lokalizációs teljesítmény javítása érdekében, klasszifikációs fázis után egy **bounding-box regressziós** nevűlépés kerül végrehajtásra, ami a megtalálandó objektumok köré úgynevezett határoló dobozokat helyez.

Azonban ennek a modelnek számos hátránya van.

* Ez egy többlépcsős modell, ahol minden egyes lépés egy független komponens.
* Az előre tanított CNN-ből kinyert jellemzőket a lemezen tárolja, hogy később betaníthassa az SVM-eket. Ehhez azonban nagy mennyiségű szabad tárhelyre van szükség.
* A Szelektív Kereső algoritmustól függ a régiójavaslatok létrehozása.
* Minden egyes régiójavasalatot függetlenül adunk át a neurális hálózatnak a jellemzők kinyeréséhez, ezért az R-CNN valós idejű futtatása lehetetlen.



**8.ábra R-CNN műkődése**

* 1. **Fast R-CNN**

A technológia fejlődésével és számítási teljesítmény növekedésével, amelyet a 2014 óta, az R-CNN tanulmány megjelenése óta tapasztalunk, most már könnyű észrevenni az R-CNN buktatóit, amiket a fentebb került kifejtésre. Ezekre a problémákra add megoldást már a Fast R-CNN, célja, hogy egy pontosabb, gyorsabb kevésbé erőforrásigényesebb megoldást nyújtson. Míg az R-CNN alapötletében minden egyes képhez körülbelül 2000 regiójavaslatot generál, és minden egyes régiójavaslatatot betáplál a mögöttes hálózati architektúrába, ez matematikailag

könnyen érthető miért is okoz próblémát, ezért a Fast R-CNN alapja, hogy a képeket csak egyszer adjuk hozzá a hálózathoz, majd a Szelektív Keresés által generált régiójavaslatokat a CNN segítségével készített Feature Map-ekre vetítjük. Ezt a folyamatot ROI (Region Of Interest) Projectionnek nevezzük. Ezután a fontos területek feltérkepézésére az SVM klasszifikálás lecserélve egy Softmax nevezetű függvényt vezettek be. Ebben az aktivációs rétegben történik az objektumok osztályainak meghatározása. A Softmax függvény célja, hogy a CNN kimenetét normalizálja, azaz nulla és egy közé sorolja be. Végül ezek a normalizált értékek megjelennek a határoló dobozokkal együtt.

A picture containing timeline

Description automatically generated

**9.ábra Fast R-CNN architektúra**

**3.3 Faster R-CNN**

A Faster R-CNN 2015-ben került publikálása, az előbbiekben ismertetett Fast R-CNN után. A Fast R-CNN már jelentős előre lépést mutatott a R-CNN architektúrák családjában, azonban a detektálás során használt szelektív keresési régiójavaslat generáló algoritmus jelentős teljesítményt igényelt. Ez az algoritmus ezért nem volt alkalmas közel valós idejű működés eléréséhez. Képenkénti ~2000 régiójavaslatot, egy új réteg bevezetésével váltották ki, ezt nevezzük Region Proposal Networknak, amely már minimális veszteséggel dolgozott. Mivel a régiójavaslatokat egy előre tanított hálózattal generálja, most már az adott probléma paramétereire lehet szabni. Így jobb régiójavaslatokat állít elő, az olyan általános módszerekhez képest, mint a Szelektív Keresés.

Timeline

Description automatically generated

**10.ábra Faster R-CNN architektúra**

**3.4 Mask R-CNN**

A Mask R-CNN a Faster R-CNN egy olyan kiterjesztése, mely felismert objektumokhoz maszkolást is társít. Működésében két fázisra bontható. Az elsőben javaslatokat generál azokról a területekről, ahol objektumok lehetnek, majd a második fázisban ezen javaslatok alapján végzi a predikciót, osztályozást, keretezést és maszkolást. Mindkét fázis a ’gerinc (backbone)’ struktúrához kapcsolódik.

A backbone egy FPN (Feature Pyramid Network – Jellemző Piramus Hálózat, lásd.: 5.ábra) és egy ResNet-ből áll. A ResNet, másnéven Residual Neural Network, azaz Maradékos Neurális Hálózat, egy olyan fajta neurális hálózat, amelyben a rétegek között engedélyezett a kihagyásos tovább ugrás. Ez csupán annyit tesz, hogy ha egy olyan réteg aktiválódik, amely egy ugrást hajt végre, akkor ez az ugrás kihagy néhány köztes réteget, ezzel orvosolva a „halványuló gradiens problémát”.

Az FPN bizonyos jellemzők kinyerésére alkalmas módszer. Szerkezete lentről-felfelé, fentről-lefelé és oldalsó irányú kapcsolatokból épül fel. Az alap koncepciója a bemeneti kép különböző méretű kiértékelésére, és ezek összehangolására épül.

Diagram

Description automatically generated

**11.ábra Feature Pyramid Network**

Az első fázisban egy RPN (Region Proposal Network – Régió Javasló Neurális Háló) neurális háló végig szkenneli a fentről-lefelé tartó irányt és javaslatokat készít azokról a területekről, ahol objektumok lehetnek. Ahhoz, hogy az eredeti képen jelezni tudjuk az objektumok feltételezett helyét, úgynevezett horgonyokra van szükség. Ezek a horgonyok igazából dobozok, előre meghatározott mérettel és elhelyezkedéssel a képen. Ezután az RPN határoló dobozokat rendel a horgonyokhoz, ezzel nagyjából meghatározva az objektumok feltételezett helyét. Azért nagyon hatékony ez a módszer, mert a jellemzők elhelyezkedése nem vész el a fel -és leméretezés során, mivel a dobozok is mérethez relatívan változnak.

A második fázisban a javasolt területek alapján egy másik neurális háló szintén végig szkenneli a feltételezett objektumokat, osztályozza, bekeretezi és maszkolja őket. Ezt azonban a meglévő horgonyok nélkül teszi, ROIAlign (Region of Interest Align) nevű módszer segítségével, ami a hasznos területek pontosítását végzi. A Mask R-CNN architektúrális felépítése a 6. ábrán látható.

Diagram

Description automatically generated

**12.ábra Mask R-CNN architektúra**

**Terv megvalósításához szükséges egyéb technológiák**

**Python**

A Python egy általános célú, nagyon magas szintű programozási nyelv. A nyelv tervezésekor az olvashatóságat és a programozói munka megkönnyítését helyezi előtérbe a futási sebességgel szemben. A Python többek között a funkcionális, az objektumorientált, az imperatív és a procedurális programozási paradigmákat támogatja. Úgynevezett interpretált nyelvek családjába tartozik, ami azt jelenti, hogy nincs különválasztva a forrás- és a tárgykód, azaz a megírt program nem futás előtt kerül gép kóddá való lefordításra, hanem futási időben értékelődik ki.

Az előre fordított nyelveknél számottevően jobb teljesítménybeli különbég figyelhető meg, mivel a fejesztő maga tudja menedzselni a hardveres erőforrásokat. Gondolok itt memória használatra, esetleg, hogy a CPU-t mennyire terhelje le. Azonban ez egyben a hátrányául is szolgál, mivel a fordítás extra időt emészt fel, ezzel szemben ez a probléma az interpretált nyelveknél nem jelentkezik. Azonban az előre fordított nyelveken írt kódok szintaxisának validálása, a fordítás pillanatában megtörténik, míg ez a másik nyelv családnál nem figyelhető meg.

Az interpretált nyelvek teljesítménybeli lemaradását az eltelt évek alatt, nagyban redukálták. Jelenleg a 3-as verziónál tart a legelőször 1991 publikált projekt. Interpretált nyelvek közül leginkább széleskörben elterjedt, ami főként az igen kiterjedt és széles körű standard könyvtárjainak köszönhet. Ezek még kiegészülnek mások által publikált nyilvános könyvtárakkal. A nyelv főbb használatos funkciók a különböző modulokban vannak. Ezeknek a moduloknak nagy részét C-ben írták meg, és beépítettek az interpreterbe, másokat Python forráskódba kell importálni.

Széleskörű könyvtár készletének és keretrendszer támogatottságának köszönhetően, ez a nyelv bizonyult eddig a legmegfelelőbbnek gépi tanulás terén. Ebből következően az objektum detektáló részt ezzel fogom elkészíteni.

**Tensorflow**

Egy nyílt forráskódú gépi tanulási keretrendszer, aminek széleskörű, flexibilis eszközei, könyvtárai vannak, lehetővé téve, hogy a fejlesztők modern gépi tanulás alapú applikációkat készítsenek. Eredetileg a Google Brain mérnöki és kutatói csapata fejlesztette ki a gépi tanulás és a mély neurális hálózatok kutatására. Azonban a rendszer elég általános, ahhoz, hogy mások által is könnyen használható legyen akik érdekeltek a témakörben. Előre tanított modellkészlettel is rendelkezik, amely jó kiindulópontként szolgálhat a témával idegeneknek, a különböző modellek megértéséhez.

További előnyét emelném ki, ami szerintem a legtöbb embernek kedvezőként szolgál. Dedikált chippeket fejlesztettek ki hozzá, amiket TPU (Tensor Processor Unit)-nak hívnak. Ezek az

egységek kifejezetten a keretrendszer által végzett műveletekre lettek specializálva. Ez mára már be is bizonyosodott, hogy ezek a speciális chip-ek jobb teljesítményt nyújtanak hagyományos fizikai társaikkal szemben.

**Google Colaboratory Pro (Colab)**

A Google Research terméke, ami lehetővé teszi, hogy bárki tetszőleges python kódot írjon és futtasson a böngészőn keresztül, és különösen alkalmas gépitanuláshoz, illetve adatelemzéshez. Technikailag a Colab egy olyan hosztolt Jupyter notebook szolgáltatás, amelynek használatához nincs szükség beállításra, ugyanakkor ingyenes hozzáférést biztosít a számítási erőforrásokhoz, beleértve a CPU-kat, GPU-kat és TPU-kat is.

**Irodalomjegyzék**

[1] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436.

[2] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learningapplied to document recognition. Proc IEEE 1998;86:2278–324.

[3] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press 2016.

[4] Maciej A Mazurowski, Mateusz Buda, Ashirbani Saha, Mustafa R Bashir Deep learning in radiology. An overview of the concepts and a survey of the state of the art with focus on MRI. 2018; 10.1002/jmri.26534.

[5] Alexander Selvikvåg Lundervold, Arvid Lundervold. An overview of deep learning in medical imaging focusing on MRI. 2018; 10.1016/j.zemedi.2018.11.002.

[6] Mingyu Kim,Jihye Yun, Yongwon Cho, Keewon Shin, Ryoungwoo Jang, Hyun-jin Bae, and Namkug Kim. Deep Learning in Medical Imaging.2019; 10.14245/ns.1938396.198.

[7] Jingzhe Ma, Shaobo Duan, Ye Zhang, Jing Wang, Zongmin Wang, Runzhi Li,Yongli Li, Lianzhong Zhang, Huimin Ma. Efficient Deep Learning Architecture for Detection and Recognition of Thyroid Nodules. 2020;10.1155/2020/1242781.

[8] Danqing Xu and Yiquan Wu. Improved YOLO-V3 with DenseNet for Multi-Scale Remote Sensing Target Detection. 2020; 10.1155/2020/1242781.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell és Jitendra Malik, „Rich feature hierarchies for accurate object detection and semantic segmentation,” 2014.

[10] Uijlings, J. R. R. et al. “Selective Search for Object Recognition.” International Journal of Computer Vision 104.2 (2013)

[11] Girshick, Ross. “Fast R-CNN.” 2015 IEEE International Conference on Computer Vision (ICCV) (2015)

|  |  |
| --- | --- |
| [12] | H. Kaiming, G. Georgia, D. Piotrs és G. Ross, „Mask-RCNN,” 2017. |