Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

Németh Marcell

Neurális háló alapú few-shot tanulás új betegségek képi felismeréséhez

TDK dolgozat

KONZULENS

Dr. Szűcs Gábor

BUDAPEST, 2020

Tartalomjegyzék

Összefoglaló

A gépi tanulás (kiemelten a deep learning) technológiák eredményes használatának szükséges, (de nem elégséges) feltétele a tanítóadatok nagy mennyiségben való rendelkezésre állása. Ennek a feltételnek számos alkalmazási területen nem lehet eleget tenni, legtöbb esetben az elérhető ismeretek hiánya vagy a szakértői tudás túlzott költségei miatt. A TDK dolgozatban bemutatott kutatási téma is egy ilyen adatokban gyakran hiányt szenvedő terület, az orvostudományban dolgozók munkáját szándékozik segíteni patológiás felvételek elemzésével, a lehető legkevesebb tanítókép felhasználásával. A választott kísérleti forgatókönyv egy manapság különösen aktuális kérdés: hogyan lehetne korábbi betegségek jellemzőmintáinak felhasználásával egy új betegséget diagnosztizálni, amelyről csak korlátozott számban állnak rendelkezésre adatok?

A kevés adatból való, azaz a „few-shot" tanulás során a modellnek az egyes osztályokból csak néhány mintát mutatunk, így az osztályozási feladat nehézségét az osztályokhoz társított, egyedi feature jellemzők minél gyorsabb és pontosabb megtanulása jelenti. Az ilyen fajta meta-tanulás során szükséges a cél képhez „hasonló" tanító képek magas szintű jellemzőinek „tudás transzfer"-szerű megtanulása. Ennek a technikának az alkalmazásával lecsökkenthető a szükséges tanítási minták száma. A feladatot eddig legjobb eredménnyel megoldani képes modellek a „prototypical network"-ök és a sziámi hálókat használó megoldások voltak. Jelen pályamunkában egy új, a korábbi megoldásoktól eltérő, továbbfejlesztett módszer került elkészítésre, amely a „Matching Network" architektúrára épül.

A dolgozat első fejezetei mélységeiben tárgyalják a „few-shot" típusú gépi tanulás hipotézistérben vizsgált elméletét és annak határait a Hilbert terekben, majd erre építve bemutatásra kerülnek a „few-shot" tanulás haladó technikái és alkalmazási lehetőségei. A dolgozatban bemutatásra kerül a továbbfejlesztett Matching Network, kiemelt figyelmet fordítva a figyelmi mechanizmusra és neurális háló architektúrára, amely képes több-nézőpontos felvételek felismerésére is. Végezetül az új modell teszt eredményei kerülnek részletezésre: a kiértékeléshez kialakított környezetben „ismeretlen" COVID-19 tüneteket mutató felvételek osztályozása pár minta felhasználásával, kizárólag más betegségek felvételein történő előzetes meta-tanulással.

Abstract

A necessary (but not sufficient) condition for the effective use of machine learning (especially deep learning) technologies is the availability of large amounts of teaching data. This condition cannot be satisfied in many applications, in most cases due to a lack of available knowledge or excessive costs of expertise. The research topic presented in the TDK dissertation is also an area that is often lacking in such data, it intends to help the work of those working in medicine by analyzing pathological recordings, using as few teaching images as possible. The chosen experimental scenario is a particularly hot issue these days: how could a new disease for which only limited data are available be diagnosed using features of previous diseases?

In the case of learning from a small amount of data, ie "few-shot", we show only a few samples of the model from each class, so the difficulty of the classification task is to learn the unique feature characteristics associated with the classes as quickly and accurately as possible. This type of meta-learning requires knowledge transfer -like learning of high-level characteristics of teaching images "similar" to the target image. By using this technique, the number of teaching patterns required can be reduced. Although there are models that can solve the problem, in the present project, a new, improved method, different from the previous solutions, was developed, which is based on the Matching Network architecture.

The first chapters of the dissertation discuss in depth the theory of few-shot type machine learning in hypothesis space and its limits in Hilbert spaces and then present the advanced techniques and application possibilities of "few-shot" learning. The dissertation presents the improved Matching Network, with a special focus on the attentional mechanism and neural network architecture, which is also capable of recognizing multi-viewpoint recordings. Finally, the results of the new model test are detailed: classification of images showing "unknown" COVID-19 symptoms in an environment designed for evaluation using a few samples, with prior meta-learning on images of other diseases only.

1 Bevezetés

A gépi tanuló technológiák hatékony alkalmazásának egyik legfontosabb feltétele a tanítóadatok nagy mennyiségben való rendelkezésre állása. Ennek a követelménynek számos alkalmazási területen nem lehet eleget tenni, legtöbb esetben az elérhető ismeretek hiánya vagy a szakértői tudás túlzott költségei miatt. A TDK dolgozatban bemutatott kutatási téma is egy ilyen adatokban gyakran hiányt szenvedő terület, az orvostudományban dolgozók munkáját szándékozik segíteni patológiás felvételek elemzésével, a lehető legkevesebb tanítókép felhasználásával.

A választott kísérleti forgatókönyv egy manapság különösen aktuális kérdés: hogyan lehetne korábbi betegségek jellemzőmintáinak felhasználásával egy új betegséget diagnosztizálni, amelyről csak korlátozott számban állnak rendelkezésre adatok? Elég belegondolni, hogy egy a COVID-19 szituációhoz hasonló világjárvány esetén a legfontosabb tényező az idő hatékony kihasználása a visszaszorításhoz. Egy új betegség felismeréséhez és a vakcina kifejlesztéséhez szükséges vizsgálatok rengeteg időt vesznek el, aminek jelentős részét az elemzésben és azonosításban kulcsszerepet játszó patológiás felvételek, azaz az adathalmaz összegyűjtése és feldolgozása teszi ki. Jelen dolgozatban egy olyan módszer kerül bemutatásra, amelynek legnagyobb erőssége a lehető legkevesebb képből való következtetések pontos levonása és ezáltal egy betegség azonosításához szükséges erőforrások minimalizálása. A képek hatékony módszerekkel való feldolgozása kulcsfontosságú a diagnózisok felállításában, ebből kifolyólag a dolgozat következő fejezeteiben az ehhez szükséges modern módszerek kerülnek bemutatásra.

1.1 ábra: Mellkasröntgen vizsgálata

forrás: www.bbva.com

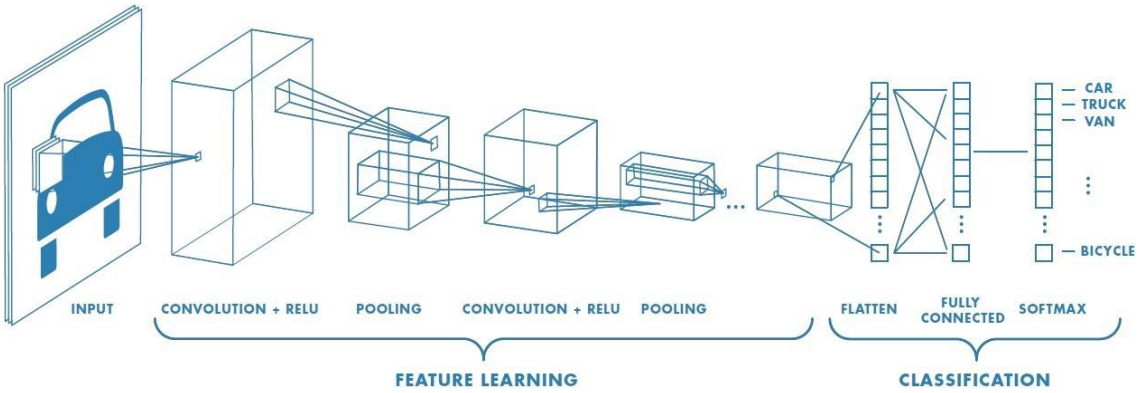
2 Elméleti összefoglaló

2.1 Neurális hálózatok

2.1.1 Konvolúciós hálózatok

Az elmúlt években a konvolúciós neurális hálók (Convolutional Neural Networks - CNN) kiemelkedő eredményeket értek el számos osztályozási és regressziós probléma esetén, főként a képi adatok osztályozásában [9].

Képek osztályozása esetén a bemenetet a felvételek pixelei adják, a kimenet leggyakrabban egy „one-hot” kódolású vektor. A gyakori sűrűn kapcsolt (dense) neurális hálók alkalmazása nem praktikus nagyobb méretű képek esetében, ugyanis a hatalmas paraméterszám könnyen túltanuláshoz vezethet. Ezzel szemben a CNN kihasználja a képi adathalmaz strukturális tulajdonságait, amik segítségével a képet és annak reprezentációit háromdimenziós (szélesség, magasság, színcsatorna) tömbként kezelheti.



2.1 ábra: CNN hálózat felépítése

forrás: mc.ai

Egy egyszerű CNN felépítéséhez minimálisan a következő rétegek szükségesek:

- A pooling réteg: célja általában a dimenziócsökkentés és a lényegkiemelés. A pooling réteget megelőző réteg kimenetét egymást nem fedő téglalap alakú (legtöbbször 2 × 2-es) részekre bontjuk. Leggyakrabban max pooling vagy average pooling réteget alkalmaznak, ahol a részekben található értékek maximuma, illetve átlaga adja a kimenetet.
- A teljes, fully-connected réteg: az *i*-edik réteg teljes, ha úgy épül fel, hogy az *i-1*-edik réteg neuronjainak mindegyike össze van kötve az *i*-edik réteg minden neuronjával.

- Konvolúciós réteg: paraméterei tanítható filterekből épülnek fel. Szélességét és magasságát tekintve minden filter kis kiterjedésű, de mélységében (ez jellemzi a használt színcsatornák számát) mindig megegyezik a bemenetével. Így az adott réteg neuronjai az előző réteg egésze helyett csak egy részével lesznek összekötve. Ezt a filtert végig vezetve a bemenet teljes magasságán és szélességén kapunk egy aktivációs térképet, ami az adott helyen mutatja a filter válaszát [8].

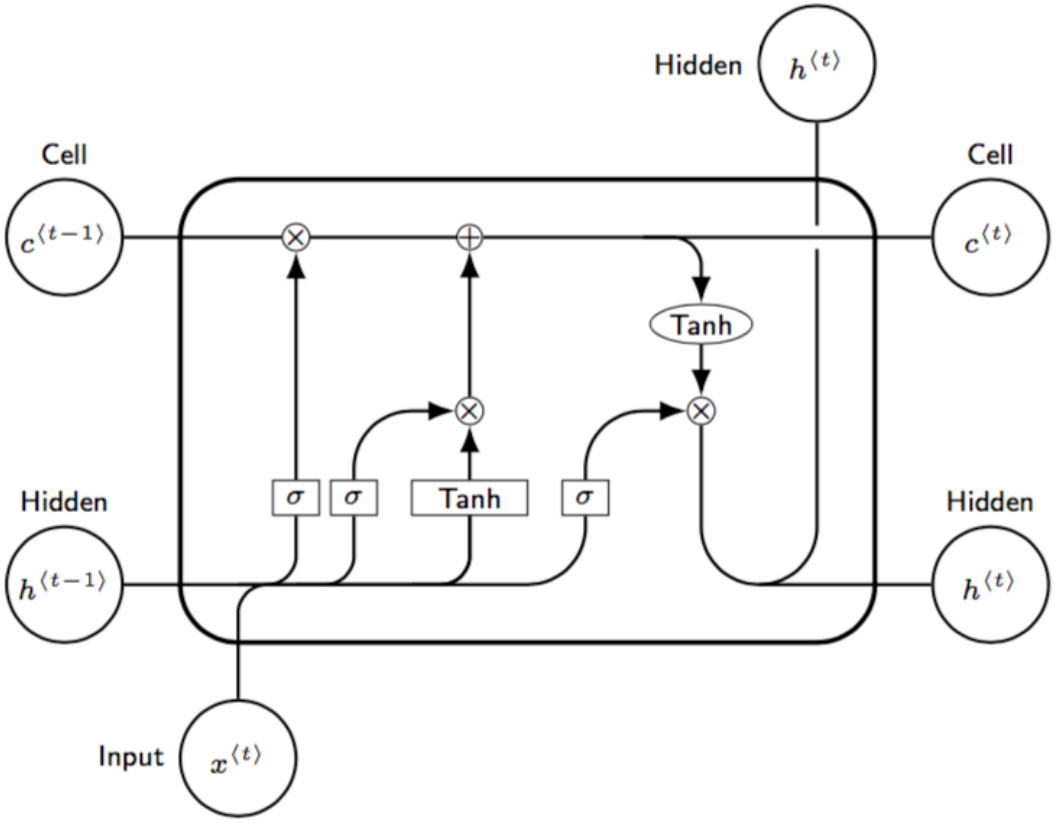
Tehát a konvolúciós réteg egy $\mathbb{R}^{W1\times H1\times D1} \rightarrow \mathbb{R}^{W2\times H2\times D2}$ leképezés, ahol a

- réteg bemenete: $W_1 \times H_1 \times D_1$ (szélesség, magasság, színcsatornák)
- paraméterei:
  - K a filterek száma, minden ponton ugyanazokat a súlyokat használva
  - F a filterek térbeli kiterjedése
  - S (stride) a lépésnagyság, ami megmutatja, hogy hány pixelt lépünk, amikor csúsztatunk
  - P (zero-padding) a kép szélét kipótoló nullák mennyisége.
- A kimenet ekkor $W_2 \times H_2 \times D_2$ lesz, ahol:
  - $W_2 = (W_1 - F + 2P)/S + 1$
  - $H_2 = (H_1 - F + 2P)/S + 1$
  - $D_2 = K$.

Az ilyen filterek képesek megtanulni bizonyos struktúrákat, mint sarkok, szélek vagy a magasabb szintű rétegekben akár méhsejt jellegű minták, illetve a lokális korrelációkat.

2.1.2 LSTM hálózatok

Amennyiben a neurális hálózat gráfreprezentációjában irányított körök találhatóak, akkor az így kapott rendszert visszacsatolt, azaz rekurrens hálózatnak (Recurrent Neural Network - RNN) nevezzük [3]. A visszacsatolt hálózatok között is többféle felépítés lehetséges, ezek csoportosíthatóak aszerint, hogy egy neuron kimenete a saját bemenetére (elemi visszacsatolás), egy másik, de azonos rétegbeli neuron bemenetére (laterális visszacsatolás) vagy pedig egy másik rétegbeli neuron bemenetére (rétegek közötti visszacsatolás) van-e kötve.



2.2 ábra: LSTM hálózat egy elemének felépítése

forrás: ai.stackexchange.com

A klasszikus perceptron modellen túl más elemi alkotóegységeket is felhasználhatunk a hálózat építése során. A visszacsatolt hálózatok közé sorolható LSTM hálózatokban az elemek felépítése bonyolultabb, mivel az elem bemenetére a bemeneti vektort és a kimeneti vektor eggyel korábbi értékét engedjük rá. Mindegyik LSTM változatra jellemző egy belső, önmagába visszacsatolt memória, melynek viselkedését ún. kapuk vezérlik. A 2.2 ábrán megfigyelhetjük az önmagába visszacsatolt memóriaegységet, illetve a kapukat is (input gate, forget gate, output gate). Minden kapu kimenete akkor aktiválódik, ha saját küszöbfüggvényének bemenete meghaladja a küszöbértéket. A bemeneti kapu (input gate) aktiválódásakor az elem bemenete (block input) bekerül a memóriaegységbe. A kimeneti kapu (output gate) aktiválásakor a memóriában tárolt érték megjelenik a neuron kimenetén [22].

A memóriaegység visszacsatolásának folyamatába a felejtő kapu (forget gate) be tud avatkozni. Ha ennek kimenete zérus, akkor az útvonalban lévő szorzás kimenete szintén zérus lesz, így a visszacsatolás megszakad, és a memóriaegység korábbi értéke elvész.

A belső memóriával rendelkező elemek felhasználása több, korábban klasszikus emlékezet nélküli visszacsatolt hálózatoknál tapasztalt problémát is megold. Az egyik

ilyen probléma, hogy a gradiens alapú hiba visszaterjesztéses tanulás során a visszaterjedő hiba grádiensének értéke drasztikusan lecsökkenhet (vanishing gradient) vagy megnőhet (exploding gradient) a bemeneti rétegek felé haladva. Ez drámaian meglassíthatja vagy tönkreteheti a bemenethez közelebbi rétegek súlyainak adaptációs folyamatát, mivel a nagy gradiens érték határozottan ki tudná jelölni a tanítás további irányát, ennek hiányában nem lesz jó a tanítás. Ezeket a problémákat az LSTM kiküszöböli.

A memória további előnye a hosszabb távú időbeli összefüggések modellezésének nagyobb lehetősége. Kapukkal szabályozott memóriaegységek esetén ugyanis a hálózat által korábban megtanult információk hosszabb ideig védett állapotban megmaradhatnak és később előhívhatóak úgy, hogy nem kell minden tanítási lépésben ezen információk felülírásától, elvesztésétől tartanunk.

2.2 Gépi tanulás kevés adatból

2.2.1 Emberi és gépi tanulás

Egy számítógépes programra akkor mondható, hogy tanulásra képes adott tapasztalatok halmazából, ha a halmazban szereplő osztályokat az egyes tanulási iterációk alatt növekvő pontossággal képes felismerni. Vegyünk példaként egy képosztályozási feladatot. A tanuló algoritmus a rendelkezésre álló tanító adatok halmazából, azaz egy orákulum által felcímkézett képek sokaságából képes megfelelő mintavételezéssel az osztályozási pontosságát fejleszteni a képekből nyert tapasztalatok alapján.

Az egyik legfontosabb különbség az emberi és gépi tanulás között a tanuláshoz szükséges minták száma. Egy embernek elég megmutatni egy-két képet bizonyos objektumokról ahhoz, hogy közel maximális pontossággal képes legyen felismerni azonos osztályba tartozó, de ismeretlen mintákat a korábbi ismereteinek felhasználásával [13], hiszen tudja, hogy melyik részlet segíthet az új minta felismerésében, azaz támaszkodik a korábbi ismereteire. Egy gépi tanuló algoritmusnak ezzel szemben ugyanennek a feladatnak a megoldásához akár több százezer minta is szükséges lehet korábbi ismeretek hiányában. Sőt, mi több, az emberihez hasonló „korábbi ismeretek” felhasználásának technikáját alkalmazó transzfertanulásnak is jelentős mennyiségű mintára van szüksége a legtöbb esetben.

A fent említett tanulási módszerek kulcsgondolatához, az általánosítás képességének megtanulásához szükséges mintaszám különbség mibenléte a legfontosabb részletek megtanulásában rejlik. Egy ember számára könnyen felismerhetőek egy adott

osztályra leginkább jellemző, vagy pedig a többitől való megkülönböztetésben leginkább hasznos tulajdonságok. A gépek ezeket az alacsony és magas szintű jellemzőket (feature) primitív módszerek használatával csak nagy mintaszám mellett képesek interpretálni, míg „okosabb", a direkt jellemzőtanulásra (meta-learning) fejlesztett tanuló algoritmusok képesek lehetnek ezt a számot lecsökkenteni akár egy-két elemre is. Fontos feladat tehát a modellek tanítását úgy megtervezni, hogy a valóban fontos részletek észrevételére legyenek képesek.

2.2.2 Few-shot tanulás

A „few-shot" tanulás (FSL – few-shot learning) [24] a gépi tanulás azon részterülete, ahol a tanítóadatok csak erősen korlátozott, alacsony számban (one-shot-nál osztályonként egy darab) állnak rendelkezésre, így a minták kevés ellenőrzött információt biztosítanak egy adott osztályról.

A legtöbb FSL feladat felügyelt tanulási problémákra vezethető vissza, az osztályozónak csak pár címkézett minta áll rendelkezésre az egyes osztályokból. A módszer leggyakoribb alkalmazási területei a képfelismerés, érzelemfelismerés szöveges adatokból, illetve az objektumok osztályozása.

A fejezetben tárgyalt problémák általános feladata egy olyan h osztályozó paraméterezése minimális számú minta felhasználásával, amely minden x_i bemenetre egy y_i címkét jósol. Az FSL szaknyelvi környezetében a tanulás során felhasznált osztályok és minták kapcsolatát leíró elnevezés az „*N-way-K-shot*" tanulás, azaz N különböző osztályból egyenként K darab mintát vételezünk a tanuláshoz, így a $D_{tanító}$ halmaz $I = K \times N$ mintával rendelkezik összesen.

Az FSL feladatok egy speciális részterülete a „zero-shot" tanulás (ZSL). Erről a problémakörről akkor beszélhetünk, ha nem áll rendelkezésre egyetlen minta sem a célosztályból a tanulás során. Ekkor az új, felismerni kívánt osztályról más modalitású ismeretek, tapasztalatok szükségesek (metaadatok), mint pl. attribútumok vagy szókörnyezetek/beágyazások egy szövegfelismerőnél.

Amikor rengeteg mennyiségű tanulóadatból történik egy gépi tanuló betanítása, több olyan modell jöhet létre a tanulás végén, amely a bemeneti mintákból képes a kimenetet előállítani (valamekkora pontossággal). Kevés tanító adat esetén viszont még sokkal nagyobb számú ilyen modell „illeszthető" a be-kimeneti párokra a sokféle lehetőség (kevés kényszer) miatt. Ezeket a modelleket felfoghatjuk egy-egy hipotézisnek, azaz egy olyan függvénynek, mely a bemenetből előállítja a kimenetet; és a függvényeket

összefogó hipotézistérben keressük a legjobb megoldást. A következő fejezetben a megoldás keresésének elméleti útját mutatom be hipotézisek terében, amely a few-shot tanulás esetén kiemelten fontos.

3 Few-shot tanulás a hipotézistérben

3.1.1 FSL tanulás módszere a hipotézisek terében

A gépi tanulás általánosságban véve a véges számú mintákból való következtetés problémájával foglalkozik. Ebben a részben bemutatásra kerül az FSL problémák alapját jelentő, felügyelt tanulás elméleti matematikai háttere és kulcs gondolatai [17].

A következőkben alkossák az $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ bemeneti minták a tanító halmazt, amely alapján egy adott szabályszerűséget megtanulva a modell képes további, ismeretlen minták *y* címkéinek becslésére, amelyeket a teszt halmazból mintavételezünk. A bemeneti mintákról továbbá elmondható, hogy egy ismert közös térbe képezhetőek le, amely a legegyszerűbb esetben egy *n* dimenziós euklideszi tér (*X*). A kimenetek címkéi analóg módon képezhetőek le egy *Y* térbe.

A fent említett szabályszerűség, azaz másnéven a keresett hipotézis egy $f: X \rightarrow Y$ függvény, amely teljesítményének számszerűsítésére az ún. empirikus hibát használjuk:

$$R_{emp}[f] = \frac{1}{m}\sum_{i=1}^{m} L(f(x_i), y_i) \tag{1}$$

ahol $L: Y \times Y \rightarrow \mathbb{R}$, a választott veszteségfüggvény. Osztályozási feladatok esetén az egyik lehetséges választás a „zero-one" veszteség:

$$L(\hat{y}, y)\begin{cases} 1, & ha\ \hat{y} \neq y \\ 0, & ha\ \hat{y} = y \end{cases} \tag{2}$$

Az eddigi jelölésekkel tehát a tanulás problémáját formalizálhatjuk egy *X* input és *Y* output térben, ahol *D* egy ismeretlen eloszlás egy $X \times Y$ térben és *F* az $f: X \rightarrow Y$ függvényeket összefogó hipotézistér, $S = (x_1, y_1), \ldots, (x_m, y_m)$ pedig *D*-ből származó minták. Ezek alapján a cél egy olyan $f \in F$ hipotézist találni, amelyre a valódi veszteség minimális:

$$R[f] = \mathbb{E}_D[L(f(x), y)] \tag{3}$$

A fenti összefüggés legnagyobb hátránya, hogy nem lehetséges minimalizálni egyértelműen, ugyanis nem ismerjük a *D* eloszlást. Erre az akadályra viszont lehetséges egy „kerülő" megoldást találni azt kihasználva, hogy a legtöbb esetben a hipotézisek valódi hibája szignifikánsan hasonló értékeket vesz fel, mint az empirikus hiba. További megfigyelés, hogy a két említett veszteség közötti eltérésben nagy szerepet játszik az *F*

hipotézistér mérete, azaz, hogy mennyire rugalmas a modellünk (mennyi szabadágfoka van). A nagyszámú szabadságfoknak viszont ára van: a „megengedő” hipotézistér velejárója a modell túltanulására való hajlam, ugyanis számtalan függvényt illeszthetünk a keresett eloszlásra. Felírva ezt a gondolatot definiálunk ún. egyenletes konvergencia határokat, amely alapján az összes *f* hipotézisre igaz egy adott hipotézistérben, hogy:

$$R[f] \leq R_{emp}[f] + \varepsilon \tag{4}$$

ahol ε az általánosítás hibája (ami általában egy összetett függvény).

Fontos megjegyezni, hogy a (4) által definiált korlátok ellenére létezhet olyan tanító halmaz összeállítás, amire a modell gyenge eredményeket produkál, emiatt a legjobb becslést, amit a várható értékre mondhatunk, a következő egyenlőtlenséggel írhatjuk fel egy adott D eloszlást tekintve:

$$\mathbb{P}\left[R[f] - R_{emp}[f] \leq \varepsilon \mid \forall f \in F\right] \geq 1 - \delta \tag{5}$$

Az ehhez hasonló egyenletes konvergencia határok megtalálása a gépi tanuló problémák legnehezebbjei között vannak számontartva. A fő nehézség abban rejlik, hogy *D* pontos ismeretének hiányában a fenti összefüggésnek fent kell állnia *D* összes lehetséges eloszlására *X*×*Y* térben (tehát nem csak egy adott *D* eloszlásra). Az (5) egyenlőtlenségnek továbbá *1- δ* valószínűséggel teljesülnie kell az összes hipotézisre szimultán is: minden f függvényre a *D* összes lehetséges eloszlásán belül felírhatjuk a fenti képletet, így kapjuk a következőt:

$$\mathbb{P}\left[R[f] - R_{emp}[f] \leq \varepsilon\right] \geq 1 - \delta \quad \forall f \in F - re \tag{6}$$

Ez utóbbi felírás azt fejezi ki, hogy bármely adott $f \in F$-re a „szerencsétlenül” mintavételezett δ hányadnyi mintákat leszámítva a (4) összefüggés igaz lesz. A (6)-os könnyebben teljesíthető, hiszen a nagyobb halmaz egészén könnyebb elérni ugyanakkora sikerességi részarányt, mintha minden részhalmazon belül el kellene érni ugyanazt az arányt. Ezzel szemben (5) előnye, hogy ennek alapján a mintavételezésből képesek vagyunk megmondani, hogy az adott tanítóhalmaz összeállítás „szerencsés” vagy pedig „szerencsétlen”. Amennyiben „szerencsés”, az egyenlőtlenség fenn fog állni az összes hipotézisre egyszerre, azaz a célunkat elértük. Ebből kiindulva célszerű az (5) egyenletet a következő alakban felírni:

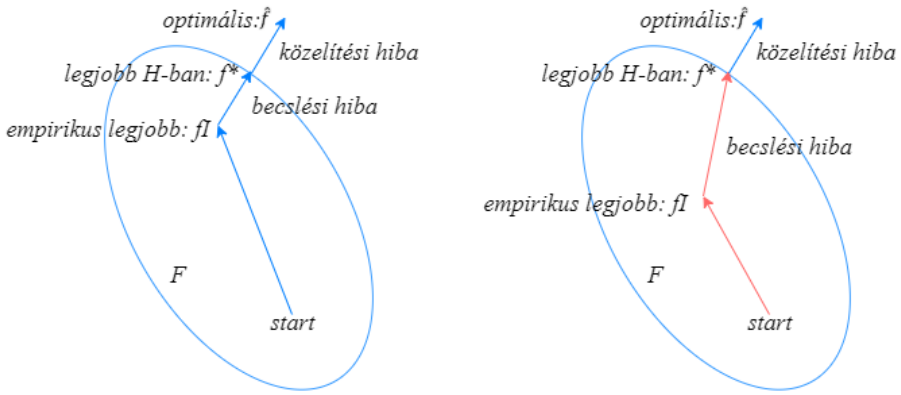
$$\mathbb{P}\left[\sup_{f \epsilon F}[\, R[f] - R_{emp}[f]\,] \leq \varepsilon\right] \geq 1 - \delta \tag{7}$$

ahol a *sup* a legalacsonyabb felső határt jelöli. A (7) és (6) között különbséget tenni meghatározó feladat abból a szempontból, hogy mire akarjuk a továbbiakban felhasználni

a határokat: az FSL tanulási probléma esetén a legfontosabb annak az *f** hipotézisnek hibája, amelyre az empirikus hiba minimális, amely viszont jelentősen függ a tanító halmaz megválasztásától. Fontos továbbá megjegyezni, hogy a tanító halmaz és az *f** hipotézis nem független véletlen változók.

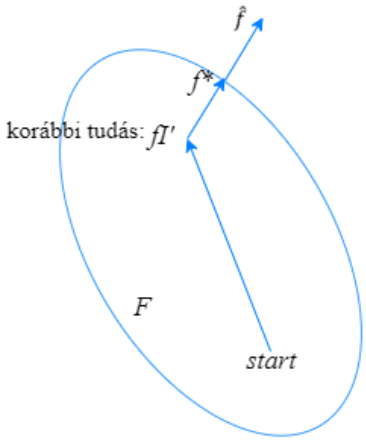
Összefoglalva elmondható, hogy amennyiben sikerül olyan határokat találni az (5) egyenlőtlenséghez, amely általánosan működőképes, a tanulás problémája megoldottnak tekinthető. Ameddig ilyen közelítés nem áll rendelkezésre, addig a bevált módszer a hipotézisek tesztelése marad az általánosító képességről árulkodó $R_{emp}[f] + \varepsilon$ hibatagra támaszkodva.

Ahogyan az az előzőekben tárgyalásra került, a modell teljes hibáját befolyásolja többek között a *F* hipotézistér és a rendelkezésre álló $D_{tanító}$ halmaz mintáinak számossága is (a mintaszám hatása a tanulás folyamatára a 3.1. ábrán látható). Ebből az állításból kiindulva a hibaminimalizálást több oldalról is meg lehet közelíteni a becslés pontatlanságának csökkentése érdekében, korábbi tudás felhasználásával [24].



3.1 ábra: Mintaszámosság hatása a tanulás folyamatára

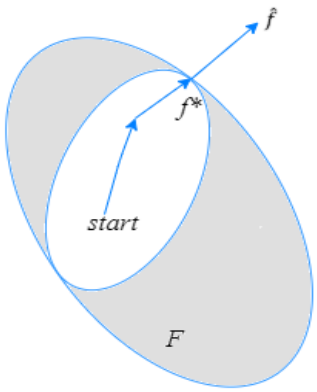
Az első ilyen megközelítés az adathalmaz felőli, amely egy „optimális” számosságú és eloszlású $D_{tanító}$ halmazt eredményez. Az ilyen módszerek a korábbi tudást felhasználva bővítik $D_{tanító}$ -t, így növelve a mintaszámot. A 3.1 ábra baloldalán látható, hogy a nagyobb mintaszámból kinyert korábbi, pontosabb tudás (f_I) felhasználásával a *start*-tal jelölt kiindulási hipotézis utáni tanulási iterációk egy, az optimális hipotézist ($\hat{f}$) *H* teren belül legjobban közelítő *f**-hoz jóval közelebb eső empirikus hipotézist eredményeznek, mint az ábra jobb oldalán látható esetben.



3.2 ábra: Adathalmaz felőli megközelítés

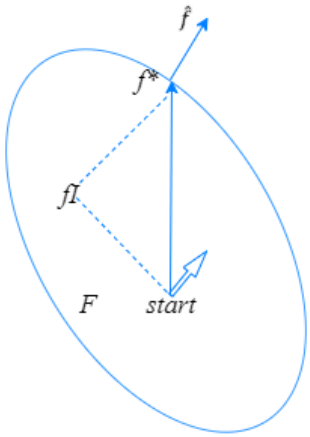
Az adathalmaz felőli megközelítésre (3.2 ábra) gyakorlati példa a tudás átvitel tanulás során (transfer learning): lényegi gondolata a magas szintű tulajdonságok, problématérre jellemző általános paraméterek tanulása egy olyan adathalmazból, ami a probléma természetét illetően nagyfokú hasonlóságot mutat (pl. mellkasröntgen felvételek, egy adott állatnak különböző fajtái) és nagy mintaszámosságú tanítóhalmazzal rendelkezik, így ennek felhasználásával sikerül *f**-hoz közel kerülni, ahogy az ábra mutatja. Egyik alterülete a domén adaptáció, aminek jellemző példája az érzelem felismerés vásárlói véleményekből egy weboldal fórumán: rendelkezésre áll számos felcímkézett hozzászólás filmekről (domén), amiken tanítjuk a modellt, de a későbbiekben nem filmekre, hanem kozmetikai cikkeket véleményező hozzászólások osztályozására fogjuk alkalmazni.

Egy másik módszer a modell részéről közelít, amely a hipotézistér meghatározásáért és szűkítéséért felelős. Ebben az esetben a priori tudás felhasználása a hipotézistér komplexitásának csökkentésére irányul, előre kizárva számos potenciális hipotézist (a 3.3 ábrán szürke terület jelöli a kizárt hipotéziseket).



3.3 ábra: Modell felőli megközelítés

A harmadik pedig az algoritmust optimalizálja: az eljárás egy optimális θ keresését végzi, amely a legjobb $f^* \in F$ hipotézist paraméterezi fel. Ezek a módszerek a priori tudás alapján egy viszonylagosan jó optimalizálási kezdőpontot definiálnak a modell számára, amelyből kiindulva hatékonyabban található meg az optimális paraméterhipotézis, ahogy az a 3.4. ábrán látható.



3.4 ábra: Algoritmus felőli megközelítés

Jelen kutatás kereteiben a FSL eredményes megvalósítását, az optimális hipotézis keresését a fenti módszerek közül a modell oldalról vizsgáljuk.

3.1.2 Hipotézisterek elmélete

Az előző fejezetben bemutatásra került az empirikus hibát minimalizáló f hipotézis keresésének bevett módszere, továbbá kifejtésre került az is, hogy az empirikus hiba minimalizálása önmagában nem elegendő, ugyanis a kizárólag ilyen fajta megközelítés túltanuláshoz vezethet. Ezt elkerülendő szükséges az F hipotézisteret szűkíteni bizonyos határok segítségével.

Egy másik módszer, a (4)-es összefüggésből kiindulva bevezet egy bűntető tagot, az $\Omega[f]$-t, amely az egyes hipotézisek komplexitásának számszerűsítését valósítja meg és az (1)-es képletben bemutatott módszer helyett a következő hibaösszeget minimalizálja:

$$R_{reg}[f] = R_{emp}[f] + \Omega[f] \tag{8}$$

ahol $\Omega[f]$ a regularizációs tag és R_{reg} pedig a regularizált hiba. Az új összefüggések bevezetésével a tanulási probléma során három összetevőre kell tehát koncentrálni: az L veszteségfüggvényre, az Ω regularizációs tagra és F hipotézistérre.

Az *F* hipotézistér konstruálása során természetes elvárás az, hogy *F* egy lineáris függvénytér legyen, amelyben bármely $f \in F$-re és λ valós számra teljesül, hogy λf is *F*-ben van, illetve bármely $f_1, f_2 \in F$-re igaz, hogy $f_1 + f_2 \in F$.

Ezeken felül *F* szerkezetének valamilyen módon kapcsolatban kell lennie az Ω regularizációs taggal. Ezt a tulajdonságot egy *F*-en definiált $\Omega[f] = || f ||^2$ normával valósítjuk meg. Az új normára nézve szintén teljesülnie kell a λ-val vett lineáris leképezéseknek, továbbá, hogy skaláris szorzatként megkapható legyen: $|| f || = \langle f, f \rangle^{1/2}$.

A fentihez hasonló skalárszorzatos vektortereket Hilbert tereknek hívjuk. A Hilbert terek lehetnek akár végtelen dimenziósak is, értelmezzük rajtuk a vetítést, lineáris transzformációkat és minden egyéb olyan műveletet is, amit az euklidészi terekben használhatunk.

A Hilbert terek nagy előnye, hogy jelen problémánkra használható Riesz Frigyes elmélete, ugyanis a Hilbert térben igaz a *Riesz reprezentációs tétel* [16], amely következtében bármely $x \in X$-re létezik egy speciális *x*-hez tartozó reprezentáns k_x függvény *H*-ban, amelyre igaz, hogy:

$$f(x) = \langle k_x, f \rangle \ \forall f \in F \tag{9}$$

A (9)-es összefüggésből nem ismerjük k_x-et, de azt biztosan tudjuk, hogy létezik. Az ötlet kulcs eleme, hogy kapcsolatot teremt *F* absztrakt szerkezete és a benne található elemek között, és bármely *x* helyett annak reprezentációját is használhatjuk. Amennyiben átírjuk a teljes regularizált hiba problémát a következőképpen:

$$\hat{\mathrm{f}} = \arg\min_{f \in F} \left[\frac{1}{m} \sum_{i=1}^{m} L\big(\langle k_{x_i}, f \rangle, y_i\big) + \langle f, f \rangle \right] \tag{10}$$

akkor látható, hogy *f* csak más függvények skaláris szorzataival vett formájában jelenik meg *F*-ben. Ebből következik, hogy ha ismerjük a skaláris szorzatot és k_x-et, akkor szabadon dolgozhatunk velük.

Ennek folytatásaként alkalmazzuk (9)-et k_x-szel valamilyen $x' \in X$-re: $k_{x'}(x) = \langle k_x, k_{x'} \rangle = k_x(x')$. Az összefüggésből látható, hogy a különböző k_x-ek belső szorzatai megmondják, hogy az egyes vektorok milyen formát vesznek fel, továbbá a szükségtelen elemeket elhagyva látható, hogy az algoritmus minőségét a belső szorzatok határozzák meg, amelyeket kerneleknek hívunk:

$$k(x, x') = \langle k_x, k_{x'} \rangle \tag{11}$$

Az egyetlen feltétel a kernelekre nézve az, hogy szimmetrikusak legyenek, valamint

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0\,, \tag{12}$$

ahol c_i, c_j valós együtthatók, és eredményezze azt, hogy $\langle \sum_{i=1}^{n} c_i k_{x_i}, \sum_{j=1}^{n} c_j k_{x_j} \rangle \geq 0$ legyen. Az ilyen tulajdonságú függvényeket pozitív definiteknek nevezzük.

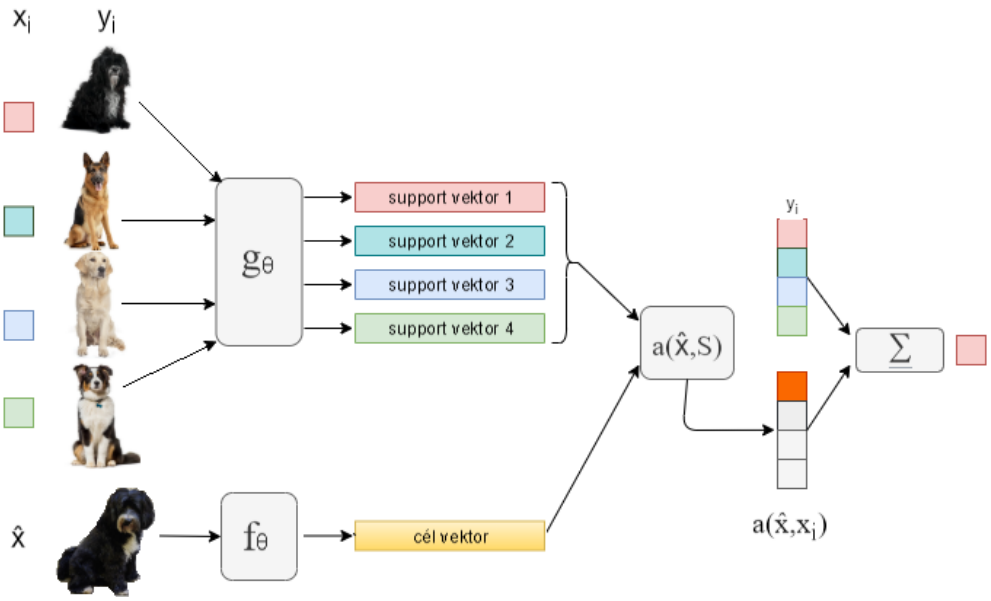
A fentieket összefoglalva elmondható, hogy a tanulás problémáját sikerült visszavezetni egy L veszteségfüggvény és k kernel megfelelő definiálására. A (10) összefüggésre ránézve látható, hogy az $\hat{f}$ a tanító adatok k_{x_1} ... k_{x_m} reprezentánsaiból fog előállni, ugyanis a hibatag csakis az f különböző k-kal vett belső szorzatától függ, míg a regularizációs tag az f függvény minden dimenziójára hatással lesz. Amennyiben f-nek van olyan komponense, ami ortogonális a k_{x_1} ... k_{x_m} által feszített altérre, úgy a hibatagra nem lesz hatással, ellenben a regularizációs tagra igen. Ebből következik, hogy az optimális f teljes valójában a reprezentánsok által feszített altérben lesz található, azaz:

$$\hat{f}(x) = b + \sum_{i=1}^{m} \alpha_i k(x_i, x) \tag{13}$$

ahol $\alpha_1 \ldots \alpha_m$ valós együtthatók és b pedig az eltolás (bias). A (13)-as képletet visszahelyettesítve (10)-be megfigyelhető, hogy a tanuló algoritmus feladata az együtthatók és az eltolás kiszámolására egyszerűsödött.

4 Matching Network architektúra

Az FSL probléma megoldására számos módszer és technikai megközelítés született, ezek közül a legjelentősebbek: *Prototypical Network* [20], *Attentive Recurrent Comparators* [18], *Simple Neural AttentIve Learner* [12], *Memory-Augmented Neural Networks (MANN)* [1], *ModelAgnostic Meta-Learning* [4] és a sziámi hálózatok [15]. Az FSL témakörében megtalálható források alapján, a mérési eredményeket elemezve és a további potenciális fejlesztéseket számba véve jelen kutatás alapjául egy újabb módszert, a *Matching Network*-öt választottuk [23], mivel ez a megoldás számos technikát adaptál, többek között a mélytanuló hálózatok által paraméterezett jellemzővektorokat felhasználó metrika [14] tanulás és a memóriával rendelkező neurális hálók eszköztárából.



4.1 ábra: Matching Network architektúra

A Matching Network típusú osztályozók lényegi gondolata, hogy kétféle tanulási módszerből építjük fel az FSL problémához a megoldást: a metrika tanulásból és a „lazy-learner" k-NN (k Nearest Neighbour, azaz k legközelebbi szomszéd) módszerből – így kutatásom is e két módszer típusra fókuszált. A metrika tanulás az elméleti bevezetőben részletezett Hilbert típusú terekben valósul meg, a terek kedvező tulajdonságait felhasználva a hipotézisfüggvények hibáinak felírásához és paraméterezéséhez, míg a k-NN típusú osztályozás az utolsó fázisban történik, a jellemzővektorok összehasonlítása során.

Az első fázisban, a röntgenfelvételek jellemzővektorainak generálásához egy később (az 5.2 fejezetben) részletezett CNN hálózatot használtam. Ennek a hálózatnak a

fő feladata egy olyan távolsági metrikát, metrika teret megtanulni, amelyben a különböző osztályokból érkező minták leképezései a lehető legjobban elkülönülnek egymástól. Az alkalmazott neurális háló feladata tehát egy ilyen tulajdonságokkal rendelkező metrika, azaz egy optimális hipotézis, függvény paraméterezése, az együtthatók kiszámolása a 3.1.2 fejezetben leírtak alapján.

Az előbbiek szerint, az első fázis minél hatékonyabb megvalósítása a második fázisban kulcsfontosságú k-NN (esetünkben 1-NN, legközelebbi szomszéd) osztályozó feladatát könnyíti meg, ugyanis a modellt nem építő (lazy learner) algoritmus a minél jobban szeparálható mintákon működik hatékonyan.

A jelen kutatásban használt Matching Network módszer két fő újítást tartalmaz a FSL megoldására. Az alkalmazott módszer minden egyes tanításhoz mintavételezett *support* halmazhoz definiál egy-egy osztályozót ($S \rightarrow C_S(\cdot)$ leképezés), majd a neurális hálók emlékező képességét kihasználva az eltárolt leképezéseket kombinálja a rendelkezésre álló tudás minél jobb felhasználása érdekében.

A fenti módszer mellett továbbá egy speciálisan FSL problémák megoldására szabott mintavételezési stratégiát használ a *support* halmazok alkalmas megválasztásával és felhasználásával.

Az elmúlt időkben számos próbálkozás történt a neurális hálózatok memóriával történő bővítésére. Szinte minden ilyen újításban egy differenciálható neurális figyelmi mechanizmus a modellek fő komponense, amely egy ún. memória mátrixot használ a már megtanult korábbi hasznos ismeretek felhasználásához.

A Matching Network típusú osztályozók legnagyobb vívmánya abban rejlik, hogy a modell tanítása után képes nagy hatékonysággal ismeretlen osztályokat is kategorizálni a hálózatban történő változtatás nélkül. A következőkben precízebben is definiáljuk a feladat leírását. Legyen egy *S support* halmaz, amely *k* darab minta/osztálycímke párost tartalmaz: $S = \{(x_i, y_i)\}_{i=1}^{k}$. Ahogyan az az 4.1 ábrán látható, a modell működése a szemléltetés kedvéért kutyafajták felismerésével lett illusztrálva röntgenfelvételek helyett. A *support* halmaz minta/címke párjait adjuk inputként egy $C_S(\hat{x})$ osztályozónak, amely egy adott $\hat{x}$ mintához egy valószínűségi eloszlást definiál $\hat{y}$ osztálycímke alapján. Ezt a leképezést matematikailag a következőképpen írhatjuk fel: legyen $S \rightarrow C_S(\hat{x}) = P(\hat{y} \mid \hat{x}, S)$, ahol a P függvényt egy neurális háló paraméterezi. Ez a konstrukció lehetővé teszi, hogy egy még nem látott mintákat tartalmazó *S' support* halmazból való osztályozás esetén módosítás nélkül fel tudjuk használni a tanítás alatt felparaméterezett modellt *S'* halmaz minden elemének osztályozásához: $\hat{x}$: $P(\hat{y} \mid \hat{x}, S')$.

Az egyes minták osztálypredikciója során használt összefüggés a következőképpen írható fel:

$$P(\hat{y}\,|\hat{x}, S) = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i \tag{14}$$

ahol x_i, y_i a minták és a hozzájuk tartozó címkék $S = \{(x_i, y_i)\}_{i=1}^{k}$ *support* halmazból és *a(·)* pedig a figyelmi mechanizmus. Érdemes észrevenni, hogy a fenti összefüggés az új osztályok mintáinak kimenetét (címkéjét) a *support* halmaz mintacímkéinek lineáris kombinációjaként állítja elő.

A figyelmi mechanizmust (*attention kernel*) alkotó modell komponensek alkalmas megválasztása kulcsfontosságú a modell hatékonyságában. Legalapvetőbb formájában a kernel a koszinusz távolságokra alkalmazott *softmax* függvény segítségével a következőképpen írható fel:

$$a(\hat{x}, x_i) = \frac{e^{c\left(f(\hat{x}), g(x_i)\right)}}{\sum_{j=1}^{k} e^{c\left(f(\hat{x}), g(x_j)\right)}} \tag{15}$$

ahol *c* függvény a koszinusz hasonlóságot írja le, f és g pedig az x_i és $\hat{x}$-ból képzett jellemzővektor elkészítéséért felelős neurális hálózatok (f és g célszerűen azonos architektúrával rendelkezik). Jelen kutatási projekt fejlesztései során a leképezések elkészítéséhez különböző architektúrájú CNN hálókat használtam fel, amelyek a 5.2 fejezetben részletesebben is bemutatásra kerülnek.

5 Továbbfejlesztett Matching Network módszer

5.1 Kutatási kérdések a kísérleti környezetben

Jelen kutatás keretei között, a Matching Network fejlesztések kísérleti környezetéhez egy igen aktuális, valós problémát tűztünk ki: olyan vírusos megbetegedések felismerését, osztályozását valósítjuk meg patológiás mellkasröntgen felvételek felhasználásával, amelyekről csak nagyon korlátozott mennyiségben állnak rendelkezésre tanítóminták. Könnyű elképzelni, hogy egy ilyen megoldás, ami képes kiterjedt adatgyűjtés és szakértői tudás (orákulum) felhasználása nélkül felismerni az új, szinte ismeretlen betegségeket, milyen lehetőségeket rejthet egy, a COVID-19 vírushoz hasonló járvány kitörésének már a kezdetén is.

A kísérleti környezet kialakításához egy nyilvánosan elérhető COVID-19 adathalmaz [2] került felhasználásra, amely kibővítésre került más betegségek mellkasröntgen felvételeivel. Fontos megjegyezni, hogy a felvételek nem rendelhetőek emberekhez, teljesen anonimizáltak, nem található a metaadataikban semmilyen páciens specifikus információ. Továbbá az adathalmaz egyes betegségeiről kétféle felvétel perspektíva – frontális és profilból (oldalról) készített felvétel – is rendelkezésre áll, ami elsőnek nehezítő körülménynek tűnt a *baseline* megoldás kialakításához, de a későbbi fejlesztések során lehetőségként tekintettünk rá. A többféle nézet kihasználását az 5.4 fejezet tárgyalja.

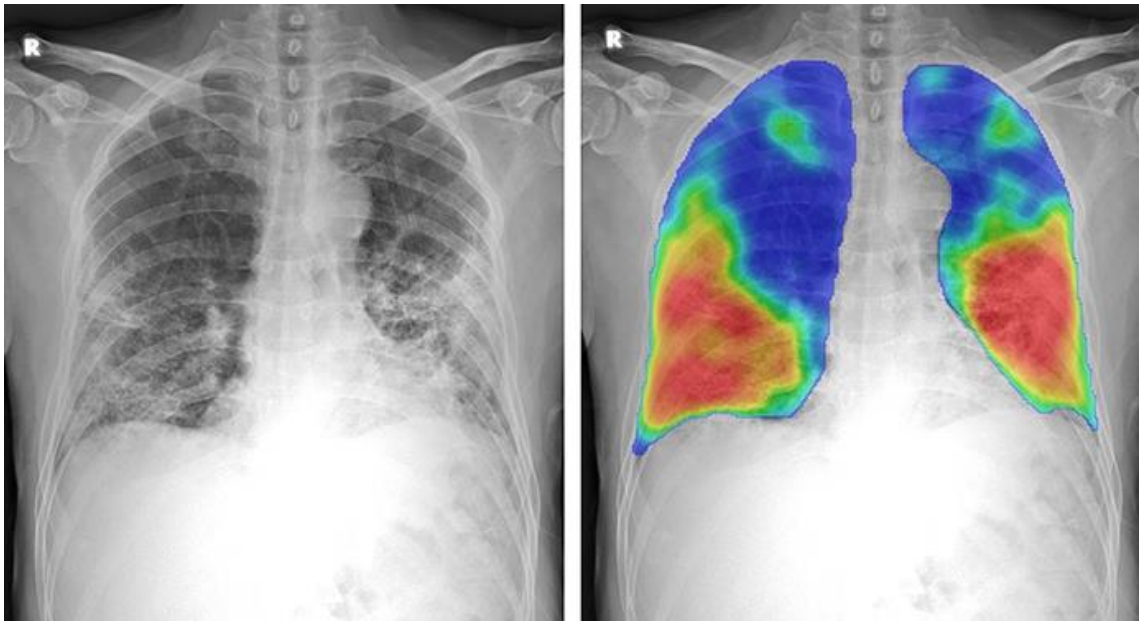
A teljes adathalmaz 758 darab felvételt tartalmaz összesen 19 betegségosztályról. Adattisztítás (hibás, vízjeles stb. felvételek eltávolítása) után 15 osztályról 680 felvétel került a végső adathalmazba.

További speciális körülmény, hogy a különböző forrásokból származó képgyűjtemények eltérő felbontással rendelkeznek, a legkisebb mindössze 150x150, míg a legnagyobbak az akár 2500x2500 pixelt is elérik. Az adathalmaz osztályait tekintve elmondható, hogy egyenlőtlen mintaszámosságot mutat az egyes felvételperspektívákat illetően:

- Teljesen kiegyenlítetlen (csak az egyik perspektívából készült képek találhatóak meg a minták között) osztályok közé tartoznak a következő betegségek: ecoli, ards, sars.
- Kiegyenlített (mindkét nézőpontból ugyanannyi kép van): influenza, mycoplasma, bacterial, chlamydophila, covid

- A többi osztályról elmondható, hogy mindkét perspektívából vannak felvételek, de nem egyenlő számban: klebsiella, legionella, lipoid, pneumocystis, pneumonia, streptococcus, varicella.

Az 5.1 ábra felvételein jól felismerhetőek a tüdő COVID-19 vírus okozta kóros elváltozásai. Bal oldal: a tünetek felismerhetőek a „sűrűbb" tüdő területekről, jobb oldal: „sűrűbb" területek hőtérképen ábrázolva.



5.1 ábra: COVID-19 tűneteket mutató mellkasröntgen felvétel.

forrás: www.delft.care

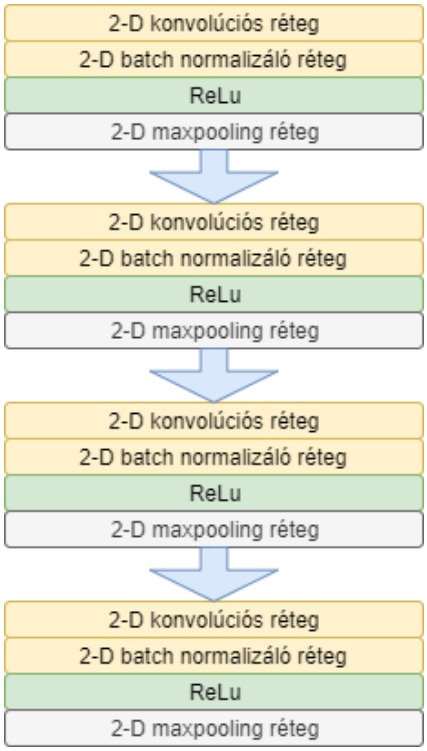
A kutatás kezdetén a következő kérdések foglalkoztattak ebben a témában:

- Mik az összefüggések a tanítóhalmaz osztályainak mintaszámai és a predikciós pontosság között? Milyen hatással van a modell teljesítményére a kiegyenlítetlen osztályok esete és hogyan kezelhető ez?
- Orvosi felvételek, jelen esetben mellkasröntgen készítésekor különböző perspektívákból készülhetnek a minták. Hogyan lehetne az egy osztályba tartozó, de különböző nézőpontból készült felvételeket hatékonyan felhasználni?
- Hogyan befolyásolja a jellemző leképezések minőségét a felhasznált neurális hálók architektúrája? Ennek a pontnak az eredménye egy „few-shot" tanuláshoz minél alkalmasabb mélytanuló hálózat, amely képes kevés iteráció segítségével is hatékonyan tanulni.

5.2 Neurális háló architektúra

A kutatás során készített modell nagyrészt már ismert, jól bevált technológiák kombinálására épül. A képek leképezéseit közös jellemzőtérbe generáló neurális hálózatok, f és g megfelelő megválasztása a modell pontosságának kulcsfontosságú összetevője. A kutatás alapcikkének tekintett [23] publikáció mindösszesen annyit osztott meg a jellemzővektorok leképezésére használt neurális architektúráról, hogy VGG [19] és Inception [21] hálózatokat használtak fel. A projekt implementációjában mi ettől a megközelítéstől eltértünk és egy sajátot alkalmaztunk, amely az 5.2 ábrán látható. Legfőbb paraméterei:

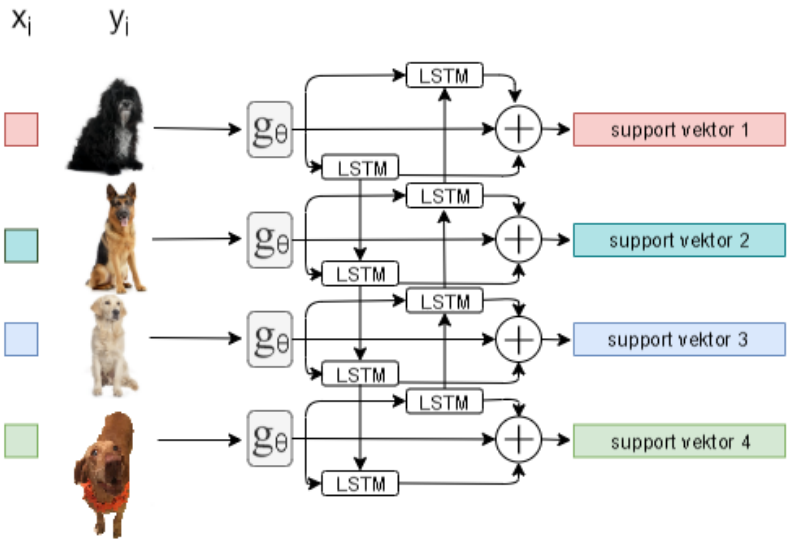
- konvolúciós réteg kernel mérete: 3×3, *stride* mérete: 1, *padding*: 1
- pooling réteg kernel mérete: 2×2, *stride* mérete: 2



5.2 ábra: CNN hálózat architektúrája

A tanítóadatok halmazából mintavételezett képek szolgálnak bemenetként a leképezéseket előállító konvolúciós hálózatoknak. A tanító iterációk során a CNN hálózatok utolsó rétegei után egy extra FC (fully connected) réteg került hozzáadásra a kimeneti vektorok előállítása érdekében. A két hálózat, f és g (tanító és teszt képek beágyazása) azonos architektúrával rendelkezik (5.2 ábra). Annak érdekében, hogy a CNN hálózat képes legyen az összes eltérő méretű képet feldolgozni, a minták egységesen 460×460 pixel nagyságra lettek középpontosan méretezve.

Fontos megjegyezni, hogy az egyes x_i minták leképezése *support* halmazonként független más mintákétól, azaz az egyes *support* halmazok nincsenek hatással egymás jellemzővektorainak leképezésére. Ezt fel lehet fogni hiányosságnak is, hiszen a minták közel azonos *doménből* származnak, egymáshoz nagyon hasonló felvételek (magasabb szinteken), így érdemes lenne kihasználni a közös kontextusukat a leképezések javításához. Praktikusabban fogalmazva, ha valamely x_i és x_j minta leképezése közel van egymáshoz a paramétertérben, akkor a jellemzővektorok pontosítása érdekében érdemes változtatni a modell paraméterein, figyelembe véve más minták leképezéseit. Ebből az ötletből kiindulva került hozzáadásra a hálózathoz egy memóriát tartalmazó komponens, a kontextus beágyazó réteg.



5.3 ábra: Kontextus beágyazó réteg

Az egyes x_i minták beágyazásához egy kétirányú *LSTM* réteg került használatra [23], amely tárolja az x_i minta *support* halmazának többi jellemzőleképezését:

$$f(\hat{x}, S) = LSTM(f'(\hat{x}), g(S), K) \tag{17}$$

ahol $f'(\hat{x})$ jelöli a CNN által generált jellemzőket, amelyek az LSTM bemeneteként szolgálnak, *g(S)* az adott *support* halmaz leképezése *g* által és *K* pedig az LSTM „időbeli léptetéseinek" száma. Ez a fejlesztés, amely számos változtatást igényelt [11] a kutatás adathalmazához való adaptálás során, az LSTM felejtő/emlékező képességét kihasználva lehetővé teszi, hogy csak bizonyos, a leképezésekhez érdemi értéket hozzáadó *support* halmazbeli elemeket hasznosítsa a figyelmi mechanizmus.

Az osztályozó f (*support* halmaz) hálózatának kontextus beágyazása a (17)-es összefüggés alapján *k* megelőző lépést feltételezve:

$$\hat{h}_k, c_k = LSTM(f'(\hat{x}), [h_{k-1}, r_{k-1}], c_{k-1}) \tag{18}$$

$$h_k = \hat{h}_k + f'(\hat{x}) \tag{19}$$

$$r_k = \sum_{n=1}^{|S|} a\big(h_{k-1}, g(x_n)\big) g(x_n) \tag{20}$$

$$a\big(h_{k-1}, g(x_n)\big) = \frac{e^{h_{k-1}^{T} g(x_n)}}{\sum_{n=1}^{|S|} h_{k-1}^{T} g(x_n)}\,, \tag{21}$$

ahol *x* a bemenet, *h* a kimenet (kimeneti kapu utáni cella) és *c* a memóriacella. Továbbá az *a* függvény a figyelmi mechanizmus a *softmax* aktivációval.

Az osztályozó *g* (cél kép) hálózatának kontextus *embeddingje*:

$$g(x_n, S) = \overrightarrow{h_n} + \overleftarrow{h_n} + g'(x_n)\,, \tag{22}$$

ahol $\vec{h}_n, \vec{c}_n = LSTM\,(g'(x_n), \vec{h}_{n-1}, \vec{c}_{n-1})$ és $\overleftarrow{h}_n, \overleftarrow{c}_n = LSTM\,(g'(x_n), \overleftarrow{h}_{n-1}, \overleftarrow{c}_{n-1})$.

5.3 Double-View Matching Network

5.3.1 Double-View Matching Network alapötlete

Ahogy korábban már olvasható volt: a mintaképek kétféle felvételi perspektívából készültek, így a mérések közben felmerült a gondolat, hogy a különböző nézőpontokat nem lenne-e érdemes valahogy elkülöníteni a modell paramétereinek finomhangolása érdekében [10][25]. Az új, két nézőpontot felhasználó megoldásom (melyet *Double-View Matching Network*-nek, röviden *DVMN*-nek neveztem el) tervezése során a legfontosabb feladat az egyes nézőpontokból készült képek jellemzővektorainak (továbbiakban vektorok) optimális felhasználása volt.

Kiindulási gondolatként felmerült, hogy az egyes nézőpontok vektorait különálló paraméterekkel rendelkező CNN hálózatoknak kell elkészíteniük a közös háló helyett [5][6]. E mögött az ötlet mögött az állt, hogy a tanítási iterációk során a kevés mintaszám miatt a modell paramétereinek megfelelő „irányba” történő hangolása kulcsfontosságú és a különböző nézőpontok felvételei könnyen félrekalibrálhatják a súlyok beállításait. Mindezek mellett a két különálló hálózat által generált leképezéseket az osztályozás előtt össze kell vonni [7], ugyanis a Matching Network több különálló vektorra való tanítás esetén a nézetek közötti különbségre tanulna rá a hasonlóságok helyett, így a megoldásom a nézetek uniójával dolgozó alapötleten alapult. A következőkben az egyes nézetek felhasználásának matematikai leírásáról lesz szó.

5.3.2 Alternatív megoldási módok az egyes nézetekhez

Az előző gondolatokat folytatva legyen S_{L1} egy címkézett képhalmaz, amely csak az első nézet képeit tartalmazza, és amelynek képeit szeretnénk egy metrikus tér betanítására használni. Ahhoz, hogy a képekből jellemzővektorokat állítsunk elő egy saját készítésű CNN-t használunk. A CNN utolsó FC rétegeit leválasztva a hálózat egy *n* elemből álló jellemzővektort állít elő minden bemeneti képhez, ezt a jellemzőkinyerő hálót *f* függvényként jelölve felírhatjuk: $v_{L1} = f_{CNN1}(x)$. L_1 címkézett képhalmaz összes képéhez ilyen módon előállított jellemzővektorok halmazát jelöljük V_{L1}-el:

$$V_{L1} = \{v_{L1} | v_{L1} = f_{CNN1}(x), x \in S_{L1}\} \tag{23}$$

Az MN háló minden beadott jellemzővektorból egy olyan új vektort állít elő, ami már az új vektortérben írja le a képet, jelöljük ezt az új vektort v'_{L1}-el, így felírhatjuk, hogy $v'_{L1} = f_{MN1}(v_{L1})$. Az így kapott új vektorok halmazát jelöljük V'_{L1}-el:

$$V'_{L1} = \{v'_{L1} | v'_{L1} = f_{MN1}(v_{L1})\} \tag{24}$$

Egy ismeretlen osztályhalmazra (ismeretlen alatt itt azt értjük, hogy az előző S_{L1} képhalmazhoz tartozó osztályok halmaza és az ismeretlen halmaz osztályainak halmaza diszjunkt halmazok, azaz metszetük üres halmaz, de az új halmazban van néhány osztálycímkével ellátott kép) szeretnénk a megtanult új vektorteret használni, ahol a képhalmaz szintén csak az első nézet képeiből áll. A korábban megtanult CNN és MN hálóval az összes képre a vektorok előállíthatók (az ismeretlen képhalmaz címkéi nélkül), így az ismeretlen képhalmazra kapott új vektorok halmazát jelöljük V'_{U1}-el, mely a következő lesz:

$$V'_{U1} = \left\{v'_{U1} \middle| v'_{U1} = f_{MN1}\left(f_{CNN1}(x)\right), x \in S_{U1}\right\} \tag{25}$$

Ha V'_{U1} elemeiből kiválasztjuk az osztálycímkével rendelkező vektorokat az ú.n. *support* halmazba (ez lesz a few-shot tanuló tanulóállománya), akkor a többi ismeretlen osztálycímkéjű vektor mindegyikét fogjuk tudni osztályozni olyan módon, hogy azt az osztálycímkét predikáljuk, amelynek a *support* vektora a legközelebb áll az osztályozandó vektorhoz.

Az előző bekezdésekben használt jelöléseket analóg módon használva a második nézőpontra:

$$V_{L2} = \{v_{L2} | v_{L2} = f_{CNN2}(x), x \in S_{L2}\}$$

$$V'_{L2} = \{v'_{L2} | v'_{L2} = f_{MN2}(v_{L2})\}$$

$$V'_{U2} = \left\{v'_{U2} \middle| v'_{U2} = f_{MN2}\left(f_{CNN2}(x)\right), x \in S_{U2}\right\}$$

A kétféle nézet felhasználása a kezdeti implementációkban egy ideálisan kialakított adathalmazon került tesztelésre: mindegyik, a tanítás és tesztelés során felhasznált osztály mintáinak *N*-számossága mindkét nézetben biztosított volt, ahol *N* a „shot”-ok számát jelöli. Ennek alapján, ha a két CNN-re az egyes nézetek képeit vezetjük, akkor azok kimenetként két *m* hosszú vektort eredményeznek, amelyeket összefűzve az egyetlen vektor előállítása érdekében, egy *2m* hosszúságú leképezést kapunk. Jelölje k_1 az első nézet, k_2 a második nézet mintáinak számát egy adott osztályban. Amennyiben az adathalmaz összeállítása a fent leírtak szerint ideális, azaz $k_1=k_2=k$, a bemeneti adattábla *k x 2m* dimenziójú lesz.

Valós környezetet szimulálva az ideális összeállítás elvárása irreális követelmény lenne, ezért ennél a pontnál a következő lehetőségek állnak rendelkezésre:

- Abban az esetben, ha legalább egy minta rendelkezésre áll mindkét nézetből, de az egyik nézetben a mintaszám nagyobb, akkor a már felhasznált mintákat újra bemenetként adhatjuk a hiányzó képek helyére. Ez a módszer könnyen túltanuláshoz vezethet a minták ismétlése miatt.
- Az előző eset körülményei állnak fent ismét, de nem szeretnénk újra felhasználni a mintákat. Ebben az esetben $min\ (k_1, k_2)$ minta kerül felhasználásra mindkét nézetből, így a bemeneti adattábla mérete $min(k_1, k_2) \times 2m$ lesz. Ennek a megoldásnak negatív hozadéka a „*shot*” szám mesterséges csökkenése és a mérések alapján várható pontosságcsökkenés.
- Nem kikötés, hogy legyen minimális mintaszám nézetenként, az egyetlen feltétel az, hogy $k_1 + k_2 \geq N$. Ez a megoldás a nézetek uniójával dolgozik és kiküszöböli a hiányos minták problémáját, az adattábla dimenziója $(k_1 + k_2) \times m$ lesz.

5.3.3 DVMN a többféle nézet kihasználására

A DVMN módszer során az előző alfejezetben vázoltakhoz hasonlóan 2 darab CNN-t tanítottam be, de az egyes vektorok összefűzése helyett a vektorok halmazainak uniójának ötletét használtam fel. Legyenek a V_{L1} és a V_{L2} a (23) egyenlettel analóg módon a jellemzővektorok halmazai. A következőkben bemutatott megoldás a nézetek uniójának kihasználására építi fel a modellt. Vegyük a jellemzővektorok unióját:

$$V_L = V_{L1} \cup V_{L2} \tag{26}$$

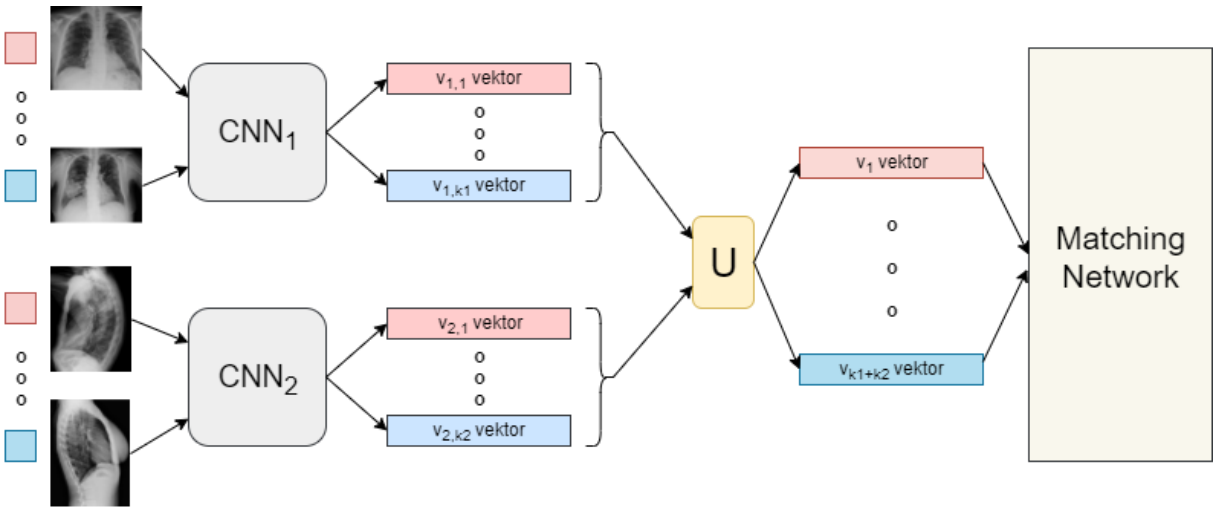
Ezt a teljes halmazt adjuk a Matching Network (MN) hálónak, hogy a few-shot osztályozáshoz szükséges vektortér tanítást el tudja végezni. Az így kapott $(k_1 + k_2) \times m$ dimenziójú új vektorok halmazát jelöljük V'_L-el:

$$V'_L = \{v'_L | v'_L = f_{MN}(v_L), v_L \in V_L\} \tag{27}$$

Amennyiben egy ismeretlen képhalmazra szeretnénk a megtanult új vektorteret használni, akkor korábban megtanult CNN_1, CNN_2 (attól függően, hogy az ismeretlen kép az első vagy második nézetbe tartozik-e) és MN hálóval az összes képre a vektorok előállíthatók (az ismeretlen képhalmaz címkéi nélkül), így az ismeretlen képhalmazra kapott új vektorok halmazát jelöljük: V'_U-val, mely a következő lesz:

$$V'_U = \left\{ v'_U \middle| \begin{matrix} v'_U = f_{MN}(f_{CNN1}(x)), x \in S_{U1} \\ v'_U = f_{MN}(f_{CNN2}(x)), x \in S_{U2} \end{matrix} \right\} \tag{28}$$

A többféle nézetet hatékonyan kihasználó és a kiegyenlítetlen osztályok problémáját kezelni képes *Double-View Matching Network* architektúrájának *support* halmaz leképező része az 5.4 ábrán látható:



5.4 ábra: Double-View Matching Network support vektor leképezése

A fejezet zárógondolataként fontos megjegyezni, hogy ugyan jelen implementáció csak kétfajta nézetet használ az adathalmaz sajátosságaiból kifolyólag, de a modell architektúrájának köszönhetően képes lenne kettő helyett N különböző perspektívát is kihasználni több halmaz uniója által.

6 Mérések és eredmények

6.1 Kezdeti osztályozók eredményei

Az újonnan fejlesztett modellek teljesítményének mérése előtt, viszonyítási alapnak egy *baseline* megoldás készült, amely modelljének egy önálló k-NN osztályozót választottam. Mérési eredmények 1-NN (legközelebbi szomszéd) jellemzővektorok összehasonlítása és osztályozása alapján adódtak az átlagolás érdekében 5 véletlenszerű teszt halmazon, ahogy ez az alábbi táblázatban látható:

Tanító/teszt osztályok száma	Pontosság (accuracy)
4 tanító – 2 teszt	0,6980
4 tanító – 4 teszt	0,6499
6 tanító – 2 teszt	0,6563
6 tanító – 4 teszt	0,6499
8 tanító – 2 teszt	0,6199
8 tanító – 4 teszt	0,5567
Átlag	*0,6384*

6.1 táblázat: k-NN osztályozó eredményei

A táblázat adatai a következő trendeket mutatják: az ismert osztályok számának növekedésével csökkenő pontosság figyelhető meg, továbbá egy osztályszám csoporton belül egyre több osztályra becslést adva monoton csökken a modell pontossága.

6.2 Tanítási tervek

Az eddigiekben bemutatásra került a Matching Network működése, amely egy-egy *support* halmazt használ egy $S \rightarrow C(\hat{x})$ osztályozó inputjaként. A módszerben a halmazonkénti mintavételezés használatával egy $P_\theta(y|\hat{x}, S)$ formájú leképezés áll elő, amelyben θ a modell paramétereit jelöli.

A tanítás során az egyes iterációkban/*epochokban*, amelyek során a gradiensek kiszámolásra kerülnek és a modell paraméterei frissülnek, elsőként mintavételezünk egy *C* osztályhalmazt *F* halmazból (összes osztály), amely az összes osztály egy részhalmazát tartalmazza, pl. {*bacterial*, *SARS*}. Következőnek *C*-t felhasználva kiválogatjuk *S*

support halmaz elemeit, egy *B batch*-csel együtt, amelyek *C* halmaz osztályainak néhány példányát tartalmazzák.

Ezután a modell paraméterei olyan módon kerülnek paraméterezésre, hogy a *B*-ben található mintákra adott osztálypredikciók hibája minimális legyen *S*-en tanítva:

$$\theta = arg\max_{\theta} E_{C\sim F}\left[E_{S\sim C,B\sim C}\left[\sum_{(x,y)\in B}\log P_{\theta}(y|x,S)\right]\right] \tag{29}$$

A batchek különböző iterációkon keresztüli mintavételezései segítenek a túltanulás elkerülésében azáltal, hogy a rendelkezésre álló képek olyan kombinációit adják bemenetként a modellnek, amikkel még nem találkozott egy adott előfordulási sorrendben. Ez a fajta megközelítés főként a kontextus beágyazásban válik előnyössé, ugyanis ismétlődő képek azonos sorozatai magukban vezethetnek túltanuláshoz az iterációkon keresztül ismétlődő (véletlent nélkülöző) sorrendjüknek köszönhetően. Amennyiben viszont nem csak a képeket, hanem kontextustanulásuk sorrendjét is variáljuk, úgy a kontextus beágyazó réteg más és más paraméterhangolásokat képes elvégezni a változó környezetnek köszönhetően.

6.3 Feladattípusokhoz tartozó tesztelési szcenáriók

A Matching Network alapú osztályozó továbbfejlesztett implementálása után szükséges volt egy széleskörű tervet készíteni. Ennek érdekében több tesztelési szcenárió készült, amelyek különböző feladattípusok vizsgálatára alkalmasak:

1. ***Új világ szcenárió***: tanulás után új osztályok osztályozásának képességét méri le.
    i. fázis: távolsági metrika tanítása *B* halmaz alapján, ahol *B* halmazban van *N* osztály
    ii. fázis: *support* halmaz (amely diszjunkt a *B* halmaztól) kiválasztása úgy, hogy *M* új osztályból osztályonként egy-egy (illetve néhány) képet választunk az osztályozóhoz
    iii. fázis: predikciók elkészítése *M* új osztályba tartozó ismeretlen képekre
2. ***Standard szcenárió***: tanulás után a megtanult osztályokon való osztályozási képességet méri le új példányokon (azaz diszjunkt teszhalmazon).
    i. fázis: távolsági metrika tanítása *B* halmaz alapján, ahol *B* halmazban van *N* osztály
    ii. fázis: *support* halmaz kiválasztása a *B* halmazból

iii. fázis: predikciók elkészítése *N* ismert osztályba tartozó, de ismeretlen képekre

3. ***Hibrid szcenárió***: tanulás után a megtanult és új osztályokon való osztályozási képességet méri le (azaz a teszhalmaz ismert és ismeretlen osztályokat is tartalmaz vegyesen).

   i. fázis: távolsági metrika tanítása *B* halmaz alapján, ahol *B* halmazban van *N* osztály

   ii. fázis: *support* halmaz kiválasztása úgy, hogy *K* (ismert és ismeretlen) osztályból osztályonként egy-egy (illetve néhány) képet választunk az osztályozóhoz

   iii. fázis: predikció elkészítése a *K* osztályba tartozó ismeretlen képekre

6.3.1 Tesztelési szcenáriók összehasonlítása

A továbbiakban ismertetem az egyes tesztelési tervek eredményeit, ahol a táblázatok a pontossági értékeket (*accuracy-t*) mutatják, azaz a helyes döntés és az összes osztályozási döntés arányát. A 1/2/5-shot tanulások tesztelését osztályonként 1/2/5 darab minta használatával végeztem, és a táblázatok első oszlopában található jelölésrendszer a következő: C<*tanító osztályok*>+C<*teszt osztályok*> */minta darab / epoch.*

A 6.2 táblázatban szereplő eredmények a korábban ismertetett adathalmazon lettek mérve a *baseline* osztályozóval (azaz itt nincs *double-view feature,* mint a *DVMN*-nél), ahol a képek között többféle nézőpontból is készült felvételek találhatóak meg.

Kiindulási mérésként a három különböző tesztfelállást szimuláló szcenárió eredményeire voltunk kíváncsiak. A következő (6.2) táblázat mérési értékei egyértelműen megmutatják, hogy már a *baseline* osztályozó is képes viszonylag jó pontossággal felismerni a teszt osztályokat, különösképpen a „Standard” szcenárió esetén teljesít jól. Az „Új világ” teszten elért eredmények biztató kiindulást adtak a projekt fő céljának kitűzött ismeretlen betegségek felismeréséhez már ismert betegségeket felhasználva.

Tanító/teszt osztályok száma	Új világ szcenárió	Standard szcenárió	Hibrid szcenárió
C4+C2/S2/E1	0,920	0,939	0,766
C4+C2/S2/E5	0,924	0,896	0,846
C4+C2/S2/E10	0,898	0,904	0,825
C4+C4/S2/E1	0,620	0,759	0,800
C4+C4/S2/E5	0,760	0,892	0,823
C4+C4/S2/E10	0,742	0,890	0,805
C6+C2/S2/E1	0,779	0,939	0,750
C6+C2/S2/E5	0,800	0,888	0,776
C6+C2/S2/E10	0,793	0,898	0,770
C6+C4/S2/E1	0,759	0,779	0,699
C6+C4/S2/E5	0,648	0,836	0,693
C6+C4/S2/E10	0,708	0,858	0,673
C8+C2/S2/E1	0,960	0,940	0,600
C8+C2/S2/E5	0,884	0,868	0,726
C8+C2/S2/E10	0,872	0,870	0,713
C8+C4/S2/E1	0,680	0,800	0,766
C8+C4/S2/E5	0,720	0,880	0,746
C8+C4/S2/E10	0,690	0,818	0,726
Átlag	0,7808	0,8696	0,7501

6.2 táblázat: Teszt szcenáriók mérési eredményei

A 6.2 táblázatban az egymás alatti 3 számérték tartozik egy méréshez oly módon, hogy az 1., 5. és 10. *epoch* utáni pontossági értékek (*accuracy*, azaz a helyesen osztályozott és az összes döntés aránya) láthatók egymás felett. A legtöbb esetben az 5. *epoch* után érte el a tanítás azt a pontossági értéket, ami után már nem tudott tovább tanulni a rendszer, ugyanis a 10. *epoch* utáni mérési eredmények (csekély mértékű

túltanulás miatt) már alacsonyabb pontosságot mutattak. Mindhárom szcenárióban átlagosan megvizsgáltam az 5. *epoch* utáni értékeket, és *0,5-5%*-os pontossági növekedést értek el az 1. *epoch*-hoz képest, majd a visszaesés miatt csak *0-3%*-os volt ez az növekmény a 10. *epoch* után. Ebből kifolyólag azt a döntést hoztam, hogy mindenhol egységesen az első 5 *epochig* tanítottam. A dolgozatom további részében bemutatott teszt eredmények is mind az első 5 *epochig* történő tanításokat tartalmazzák (így ezt külön már nem tüntettem fel).

6.3.2 Új betegségek osztályozásának eredményei

A továbbiakban a dolgozat fő célkitűzését adó téma, az új betegségek felismerését szimuláló „Új világ" szcenárió mérései következnek. A táblázatokban az új osztályozók (MN - *baseline*, és DVMN) kerülnek összehasonlításra egy paraméter hangolást nem alkalmazó, Matching Network komponenst nem tartalmazó 1 nézetes k-NN osztályozóval, illetve a DVMN oszlopokban található *2-view k-NN*-nel, amely két perspektíva képeivel is dolgozik. Az adathalmaz szétválasztása, azaz a két különböző nézőpontból készült felvételek különválogatása utáni mérési eredmények a következőképpen alakultak:

A 6.3 táblázat eredményeire nézve látható, hogy a one-shot-learning feladatnál már egyetlen minta felhasználásával is az új *DVMN* módszer teljesít a legjobban ***81,2%*** elért átlagos pontossággal. Az osztályozók két külön nézeten mért részeredményei a Függelék 1. táblázatában találhatóak.

	Külön nézetek átlaga		DVMN		Kevert nézetek	
	2 MN átlaga	*2 k-NN átlaga*	*DVMN*	*2-view k-NN*	*MN*	*k-NN*
C4+C2/S1	**0,8566**	0,6560	0,7766	0,6200	0,8380	0,5833
C4+C4/S1	**0,8683**	0,6333	0,8333	0,6366	0,7640	0,6500
C6+C2/S1	0,8149	0,5933	**0,8200**	0,6600	0,7940	0,6499
C6+C4/S1	0,7599	0,5426	**0,7600**	0,5666	0,7240	0,5400
C8+C2/S1	0,7450	0,6205	0,8333	0,6000	**0,8740**	0,5600
C8+C4/S1	0,7900	0,6220	**0,8466**	0,5200	0,7700	0,6199
Átlag	0,8057	0,6112	**0,8116**	0,6005	0,794	0,6005

6.3 táblázat: Új világ szcenárió eredményei osztályonként 1 darab mintával

A 6.4 táblázatból kiolvasható, hogy a mintaszám növelése esetén, 2-shot mintavételezésnél a tesztesetek döntő többségében a DVMN implementáció éri el a legjobb osztályozási teljesítményt ***85,7%***-os átlagos pontossággal. Az osztályozók két külön nézeten mért részeredményei a Függelék 2. táblázatában találhatóak.

	Külön nézetek átlaga		DVMN		Kevert nézetek	
	2 MN átlaga	*2 k-NN átlaga*	*DVMN*	*2-view k-NN*	*MN*	*k-NN*
C4+C2/S2	0,9000	0,6828	0,8633	0,6300	**0,924**	0,6980
C4+C4/S2	**0,9000**	0,6616	0,8434	0,5833	0,760	0,6499
C6+C2/S2	0,8459	0,6425	**0,8566**	0,6166	0,800	0,6563
C6+C4/S2	0,7680	0,6649	**0,7966**	0,6500	0,678	0,6499
C8+C2/S2	0,7739	0,6636	**0,9133**	0,5333	0,884	0,6199
C8+C4/S2	0,7340	0,6499	**0,8666**	0,5833	0,720	0,5567
Átlag	0,8203	0,6608	**0,8566**	0,5994	0,7943	0,6384

6.4 táblázat: Új világ szcenárió eredményei osztályonként 2 darab mintával

A 6.5 ábrán észrevehető, hogy az 5 mintát használó mérési eredmények legfeljebb 8 külön osztályig kerültek rögzítésre. Ennek oka, hogy a második nézetből jelentősen kevesebb minta áll rendelkezésre, mint az elsőből, így az eredmények összehasonlítása statisztikailag nem lett volna biztosítható. Az előző két méréshez (6.3 és 6.4 táblázatok) képest ugyan kevesebb teszteset állt rendelkezésre, de a 6.5 táblázat eredményei alapján a DVMN osztályozó teljesítménye bizonyult a legjobbnak ebben az esetben is ***85,4%***-os átlagos pontossággal.

	Első nézet		DVMN		Kevert nézetek	
	MN	*k-NN*	*DVMN*	*2-view k-NN*	*MN*	*k-NN*
C4+C2/S5	0,8180	0,6333	**0,8433**	0,6400	0,7780	0,6100
C4+C4/S5	0,7580	0,5800	0,7960	0,5200	**0,8159**	0,6400
C6+C2/S5	0,8320	0,6599	**0,924**	0,4400	0,8539	0,6333
Átlag	0,8026	0,6244	**0,8544**	0,533	0,8159	0,6277

6.5 táblázat: Új világ szcenárió eredményei osztályonként 5 darab mintával

7 Összefoglalás

A dolgozatban részletezésre került a kevés adatból történő gépi tanulás elméleti háttere, kiemelve a hipotézisterekben történő tanuló folyamatok technikáit és lehetőségeit. A matematikai háttér bemutatását követően egy aktuális gyakorlati problémán - új / ismeretlen betegségek mellkasröntgen felvételeinek elemzésén - keresztül került vizsgáltam egy új FSL technológiát, amely a Matching Network osztályozóra épül. A vizsgálatom középpontjában az alap Matching Network architektúra teljesítményének felmérése és továbbfejlesztése állt. Elsőként a *baseline* implementáció teljesen új adathalmazon (röntgenfelvételek az eredeti *Imagenet* halmaz helyett) történő méréseit végeztem el, majd a teljesítményprofil felállítása után a lehetséges fejlesztési ötletek megvalósítása következett.

A kutatás első felében elkészült egy olyan továbbfejlesztett Matching Network módszer, amely képes különböző felbontású mellkasröntgen felvételek hatékony osztályozására három különféle teszt szcenárióban is. A munka későbbi fázisaiban kiválasztásra került egy kiemelt gyakorlati potenciállal rendelkező teszteset a három közül, amely ismeretlen betegségeket képes osztályozni kizárólag más betegségek jellemzőinek felhasználásával mindössze néhány minta segítségével.

A kutatás második felének legfontosabb fejlesztése a *Double-View Matching Network (DVMN)* neurális architektúra és osztályozó létrehozása volt, amely képes páciensfüggetlen, kétfajta nézőpontból készített felvételek hatékony jellemzőleképezésére és rajtuk keresztül a betegségek felismerésére új metrikatanuló módszerek felhasználásával. Továbbá az új osztályozó tervezése során olyan módszerek kerültek beépítésre a modellbe, mint például egy új kontextus beágyazó réteg, amely a tanító batch-ek mintáinak közös jellemzőit kihasználva teszi hatékonyabbá a jellemzőleképezéseket az iterációk során.

A munka végső fázisában a különféle szcenáriókat és azokon belül számos tesztesetet kipróbálva kerültek rögzítésre az új modellek teljesítményei. Az eredmények azt mutatják, hogy a kutatási projekt céljául szolgáló ismeretlen betegségek felismerésének feladatát kiemelkedő hatékonysággal sikerült megvalósítani nehéz körülmények (kevés minta, páciensfüggetlen képek stb.) között is.

A projekt megvalósítása során nem csak új eredményeket sikerült elérni, hanem számos kérdés is összegyűlt. Ezek közül kiemelt jelentőségű téma lehet későbbi fejlesztésekhez a neurális architektúra felépítésének további vizsgálata, mint például

autoencoderek és más hálózati építőelemek használata, illetve olyan több nézőpontos módszerrel való kibővítési lehetőség is, amely ha egyszerre rendelkezésre állnának ugyanarról a páciensről különböző perspektívájú képek, akkor azt ki tudná használni. Jelenlegi adathalmaz ugyanis olyan, hogy van ugyan 2 féle nézőpont a képhalmazban, de minden vizsgálati alanyról csak az egyik vagy a másik áll rendelkezésre.

A dolgozatban tárgyalt orvosi képfeldolgozáson kívül az FSL néhány gyakoribb felhasználása a még következő területeken lehetséges:

- Az emberi tanulás minél valósabb szimulálása: a gépi gondolkodás minél „emberibb" megvalósítása érdekében fontos mérföldkőnek számít az FSL eredményes kivitelezése. Az egyik leggyakoribb tesztfeladat egy új karakter generálása mindösszesen pár darab előre elkészített minta felhasználásával. Az emberi gondolkodás mintájára a modell felhasználja az adott mintákból gyűjtött tapasztalatokat akár más, előre tanított összefüggésekkel együtt (ilyen összefüggés lehet például részletek, és egymáshoz való kapcsolatuk). A kimenetként képzett új karakterek végül egy vizuális Turing teszten esnek át, amely eldönti, hogy ember vagy pedig gép készítette őket. Ezekkel a korábbi tapasztalatokkal és tudással felvértezve a gép képes osztályozási feladatok ellátására is.
- Ritka esetek problémája: a mindennapok során, nem szimulált „tökéletes környezetben" csak elvétve állnak rendelkezésre elegendő mennyiségű és minőségű tanító halmazok. Az FSL megfelelő használata pontosan az ilyen esetekre lett kitalálva. Jelen dolgozat kísérleti tesztkörnyezete is ezt a felhasználási területet szimulálja: a COVID-19 vírust szándékozik felismerni mellkasröntgen felvételeken alig pár darab mintából, más tüdőbetegségek jellemzőtereinek további paraméterezésével, felhasználásával.
- Adatgyűjtés és számítási erőforrás csökkentése: az FSL megoldások hatékony megvalósítása és használata jelentős mennyiségű szakértői tudás igénybevételét képes helyettesíteni a paraméterterek optimális tanulásával. A kevés tanító adat feldolgozása kevesebb számítási kapacitást is igényel, továbbá a modellek nagyfokú általánosítóképességüknek köszönhetően sok hasonló problémakörben újra felhasználhatóak.

Köszönetnyilvánítás

A TDK dolgozatban ismertetett eredmények a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Kar Balatonfüredi Hallgatói Kutatócsoport szakmai közössége keretében jöttek létre a régió gazdasági fejlődésének elősegítése érdekében. Az eredmények létrehozása során figyelembe vettük a balatonfüredi központú Rendszertudományi Innovációs Klaszter által megfogalmazott célkitűzéseket, valamint a párhuzamosan megvalósuló EFOP 4.2.1-16-2017-00021 pályázat támogatásával elnyert „BME Balatonfüredi Tudáscentrum” térségfejlesztési terveit.

A kutatás az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg (EFOP-3.6.2-16-2017-00013, Innovatív Informatikai és Infokommunikációs Megoldásokat Megalapozó Tematikus Kutatási Együttműködések).

Irodalomjegyzék

[1] Collier, M., & Beel, J. (2019). Memory-Augmented Neural Networks for Machine Translation. ArXiv, abs/1909.08314. https://arxiv.org/abs/1909.08314

[2] COVID-19 adathalmaz elérhetősége: https://github.com/ieee8023/covid-chestxray-dataset

[3] De Mulder, Wim & Bethard, Steven & Moens, Marie-Francine. (2014). A Survey on the Application of Recurrent Neural Networks to Statistical Language Modeling. Computer Speech & Language. 30. 10.1016/j.csl.2014.09.005. https://www.researchgate.net/publication/266204519_A_Survey_on_the_Application_of_Recurrent_Neural_Networks_to_Statistical_Language_Modeling

[4] Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ArXiv, abs/1703.03400. https://arxiv.org/abs/1703.03400

[5] Geras, Krzysztof & Wolfson, Stacey & Kim, S. & Moy, Linda & Cho, Kyunghyun. (2017). High-Resolution Breast Cancer Screening with Multi-View Deep Convolutional Neural Networks. https://www.researchgate.net/publication/315492554_High-Resolution_Breast_Cancer_Screening_with_Multi-View_Deep_Convolutional_Neural_Networks

[6] Image Segmentation. 8535-8545. 10.1109/CVPR.2019.00874. https://www.researchgate.net/publication/338511849_Data_Augmentation_Using_Learned_Transformations_for_One-Shot_Medical_Image_Segmentation

[7] Kan M, Shan S. and Chen X., "Multi-view Deep Network for Cross-View Classification," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 4847-4855, doi: 10.1109/CVPR.2016.524. https://ieeexplore.ieee.org/document/7780893

[8] Katona: Mélytanulási módszerek az orvosi képalkotó diagnosztikában http://web.cs.elte.hu/blobs/diplomamunkak/bsc_matelem/2018/katona_reka.pdf

[9] Li, Z., Yang, W., Peng, S., & Liu, F. (2020). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *ArXiv, abs/2004.02806*. https://arxiv.org/abs/2004.02806

[10] Li,Y., Yang M. and Zhang Z., "A Survey of Multi-View Representation Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 10, pp. 1863-1883, 1 Oct. 2019, doi: 10.1109/TKDE.2018.2872063. https://ieeexplore.ieee.org/document/8471216

[11] Matching Network – a felhasznált kiindulási implementáció elérhetősége: https://github.com/gitabcworld/MatchingNetworks

[12] Mishra, N., Rohaninejad, M., Chen, X., & Abbeel, P. (2017). Meta-Learning with Temporal Convolutions. ArXiv, abs/1707.03141. https://arxiv.org/abs/1707.03141

[13] Mohsen Kaboli. A Review of Transfer Learning Algorithms. [Research Report] Technische Universität München. 2017. ffhal-01575126
https://hal.archives-ouvertes.fr/hal-01575126/document

[14] Parag:Metric learning tutorial
https://parajain.github.io/metric_learning_tutorial

[15] Ramachandra, B., Jones, M.J., & Vatsavai, R. (2020). Learning a distance function with a Siamese network to localize anomalies in videos. 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), 2587-2596.
https://arxiv.org/abs/2001.09189

[16] Riesz Reprezentációs Tétel:
http://www.math.uwaterloo.ca/~beforres/PMath451/Course_Notes/Chapter6.pdf

[17] Risi: A Short Introduction to Hilbert Space Methods in Machine Learning
http://www.cs.columbia.edu/~risi/notes/tutorial6772.pdf

[18] Shyam, P., Gupta, S., & Dukkipati, A. (2017). Attentive Recurrent Comparators. ArXiv, abs/1703.00767.
https://arxiv.org/abs/1703.00767

[19] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.
https://arxiv.org/abs/1409.1556

[20] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical Networks for Few-shot Learning. ArXiv, abs/1703.05175.
https://arxiv.org/abs/1703.05175

[21] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9.
https://arxiv.org/abs/1409.4842

[22] Szórádi: A beszéd alapfrekvenciájának statisztikai vizsgálata
https://diplomaterv.vik.bme.hu/hu/Theses/A-beszed-alapfrekvenciajanak-statisztikai

[23] Vinyals et al (2016): Matching Networks for One Shot Learning
https://papers.nips.cc/paper/6385-matching-networks-for-one-shot-learning.pdf

[24] Wang, Yaqing & Yao, Quanming. (2019). Generalizing from a Few Examples: A Survey on Few-Shot Learning
https://www.researchgate.net/publication/332342190_Generalizing_from_a_Few_Examples_A_Survey_on_Few-Shot_Learning

[25] Zhao, A., Balakrishnan, G., Durand, F., Guttag, J. V., & Dalca, A. V. (2019). Data augmentation using learned transformations for one-shot medical image segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8543-8553).
https://arxiv.org/abs/1902.09383

Függelék

1-shot teszteset részletes mérési eredményei:

	Első nézet		*Második nézet*		*Nézetek átlaga*	
	MN	*k-NN*	*MN*	*k-NN*	*2 nézet átlaga (MN)*	*2 nézet átlaga (k-NN)*
C4+C2/S1	0,8866	0,6788	0,8266	0,6333	0,8566	0,6560
C4+C4/S1	0,8433	0,6500	0,8933	0,6166	0,8683	0,6333
C6+C2/S1	0,8233	0,5400	0,8066	0,6466	0,8149	0,5933
C6+C4/S1	0,7966	0,5687	0,7233	0,5166	0,7599	0,5426
C8+C2/S1	0,7467	0,6778	0,7433	0,5633	0,7450	0,6205
C8+C4/S1	0,8100	0,6275	0,7700	0,6166	0,7900	0,6220

Függelék 1. táblázat: 1-shot mérési eredmények a két nézetre

2-shot teszteset részletes mérési eredményei:

	Első nézet		*Második nézet*		*Nézetek átlaga*	
	MN	*k-NN*	*MN*	*k-NN*	*2 nézet átlaga (MN)*	*2 nézet átlaga (k-NN)*
C4+C2/S2	0,9360	0,7000	0,8640	0,6657	0,9000	0,6828
C4+C4/S2	0,8960	0,6771	0,9040	0,6461	0,9000	0,6616
C6+C2/S2	0,8760	0,6500	0,8159	0,6350	0,8459	0,6425
C6+C4/S2	0,7760	0,7000	0,7600	0,6299	0,7680	0,6649
C8+C2/S2	0,7759	0,7272	0,7720	0,6000	0,7739	0,6636
C8+C4/S2	0,7280	0,6499	0,7400	0,6499	0,7340	0,6499

Függelék 2. táblázat: 2-shot mérési eredmények a két nézetre