

Új betegségek képi osztályozását segítő few-shot tanulás mély neurális hálózattal

Szűcs Gábor¹, Németh Marcell²

¹ Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék,
Budapest

² Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Balatonfüredi Hallgatói Kutatócsoport
{szucs@tmit.bme.hu, marcell.nemeth@edu.bme.hu}

Absztrakt. A gépi tanulás egyik jellemzően szükséges feltétele a tanítóadatok nagy mennyiségben való rendelkezésre állása; ennek a feltételnek azonban számos területen nem lehet eleget tenni, a szakértői tudás vagy pedig a rendelkezésre álló források hiánya miatt. Cikkünk a kevés mintából történő tanulással, az ún. few-shot tanulással foglalkozik, melynek során az egyes osztályokból csak néhány mintát mutatunk (maximum 5). Az ilyen fajta meta-tanulás során szükséges a cél képekhez hasonló tanító képek magas szintű jellemzőinek „tudás transzfer”-szerű megtanulása. Cikkünkben egy új, továbbfejlesztett módszert javasolunk, amely a „Matching Network” architektúrára épül. Módszerünk a mély neurális hálónál alkalmazott figyelmi mechanizmust használja és a neurális háló architektúrája úgy lett kialakítva, hogy képes különböző nézőpontos felvételek felismerésére is. Az új módszer kiértékeléséhez kialakított környezetben „ismeretlen” COVID-19 tüneteket mutató felvételek osztályozása volt a feladat pár minta felhasználásával, kizárólag más betegségek felvételein történő előzetes meta-tanulással. A teszt eredményeink azt mutatták, hogy az általunk javasolt DVMN módszer érte el a legnagyobb pontosságot.

Kulcsszavak: few-shot tanulás, mély neurális hálózat, COVID-19, képosztályozás, különböző nézőpontos felvételek

1 Bevezetés

A „few-shot” tanulás (FSL – few-shot learning) [15] a gépi tanulás azon részterülete, ahol a tanítóadatok csak erősen korlátozott, alacsony számban állnak rendelkezésre. Mivel ezek kevés információt biztosítanak egy adott osztályról (az osztályozási feladatnál), így az FSL során szükség van egy olyan másik adathalmazra, ahol nem kell ugyanolyan osztályoknak lenniük, de elvárt, hogy hasonló jellegű legyen, hogy azoknak az adatoknak a magas szintű jellemzőinek „tudás transzfer”-szerű megtanulásával egyfajta meta-tanulás megvalósítható legyen. A legtöbb FSL feladat felügyelt tanulási problémákra vezethető vissza, ennek leggyakoribb

alkalmazási területei a képfelismerés, érzelemfelismerés szöveges adatokból, illetve az objektumok osztályozása. 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 (one-shot learning esetében osztályonként csak 1 darab mintát) veszünk a tanuláshoz, így a $D_{\text{tanító}}$ halmaz $I = K \times N$ mintával rendelkezik összesen.

Az FSL probléma megoldására számos módszer és technikai megközelítés született, ezek közül a legjelentősebbek: *Prototypical Network* [12], *Attentive Recurrent Comparators* [10], *Simple Neural Attentive Learner* [7], *Memory-Augmented Neural Networks (MANN)* [1], *ModelAgnostic Meta-Learning* [3] és a szími hálózatok [9]. Az FSL témakörében megtalálható források alapján, a mérési eredményeket elemezve és a további potenciális fejlesztéseket számba véve kutatásunk alapjául a *Matching Network* módszert választottuk [14], mivel ez a megoldás számos technikát adaptál, többek között a mélytanuló hálózatok által paraméterezett jellemzővektorokat felhasználó metrika tanulás [8] és a memóriával rendelkező neurális hálók eszköztárából. Így a következőkben ezt a módszert ismertetjük, majd bemutatjuk az általunk javasolt, továbbfejlesztett módszert, mely különböző nézőpontból készített képek felismerésére is alkalmas.

2 Matching Network architektúra

A Matching Network típusú osztályozók lényegi gondolata, hogy kétféle tanulási módszerből építi fel az FSL problémához a megoldást: a metrika tanulásból (első fázis) és a k -NN osztályozóból (k Nearest Neighbour, azaz k legközelebbi szomszéd módszer) a második fázisban.

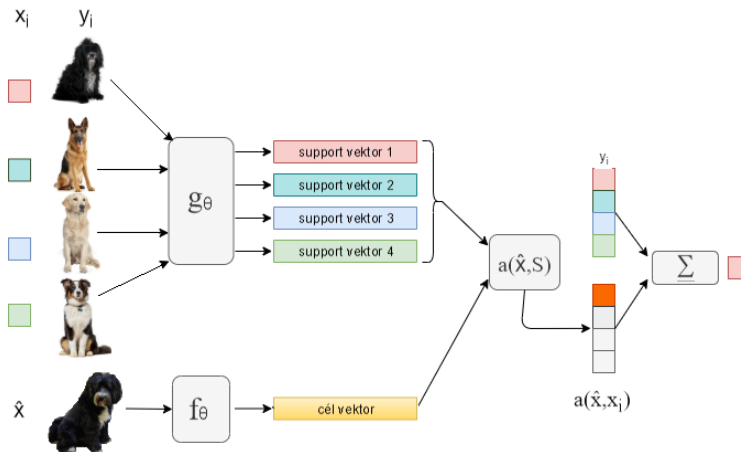


Fig. 1. Matching Network architektúra

Az első fázisban a jellemzővektorokat kell előállítani, ezt egy mély neurális hálózattal lehet megtenni; 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 előbbiek szerint, az első fázis minél hatékonyabb megvalósítása a második fázisban kulcsfontosságú k -NN (esetünkben 1-NN, azaz legközelebbi szomszéd) osztályozó feladatát könnyíti meg, ugyanis a modellt nem építő (lusta tanuló) algoritmus a minél jobban szeparálható mintákon működik hatékonyan. Végül a Matching Network típusú osztályozók a modell tanítása után képesek lesznek ismeretlen osztályokat is kategorizálni a modellben történő változtatás nélkül.

A Matching Network módszer minden egyes tanításhoz mintavételezett *support* halmazhoz definiál egy-egy osztályozót ($S \rightarrow C_s(\cdot)$ leképezés), majd a neurális háló emlékező képességét (LSTM típusú neurális háló esetében van ilyen memóriája a hálózatnak) kihasználva az eltárolt leképezéseket kombinálja a rendelkezésre álló tudás minél jobb felhasználása érdekében. A memóriára a figyelmi mechanizmus miatt van szükség, ugyanis ez egy olyan komponens, amely egy ún. memória mátrixot használ a már megtanult korábbi hasznos ismeretek felhasználásához.

A következőkben a fentieket formálisabb módon is bemutatjuk. Legyen egy S *support* halmaz, amely k darab minta/osztálycímke párost tartalmaz: $S = \{(x_i, y_i)\}_{i=1}^k$. Ahogyan a Fig. 1 ábrán látható, a modell működése a szemléltetés kedvéért kutyafajták felismerésével lett illusztrálva (a később ismertetet röntgenfelvételek helyett). A *support* halmaz minta/címke párait 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 $S \rightarrow C_s(\hat{x}) = P(\hat{y} | \hat{x}, S)$, 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: $\hat{x}: P(\hat{y} | \hat{x}, S')$.

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

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

ahol x_i, y_i a minták és a hozzájuk tartozó címkék $S = \{(x_i, y_i)\}_{i=1}^k$ *support* halmazból és $a(\cdot)$ pedig a figyelmi mechanizmus. A fenti összefüggés az új osztályok mintáinak kimenetét (címkéjét) tehát 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:

$$a(\hat{x}, x_i) = \frac{e^{c(f(\hat{x}), g(x_i))}}{\sum_{j=1}^k e^{c(f(\hat{x}), g(x_j))}} \quad (2)$$

ahol c függvény a koszinusz hasonlóságot írja le, f és g pedig az x_i és \hat{x} -ből képzett jellemzővektor elkészítéséért felelős neurális hálózatok (f és g célszerűen azonos architektúrával rendelkeznek).

2 Továbbfejlesztett Matching Network röntgenfelvételekhez

A Matching Network továbbfejlesztését egy igen aktuális, valós probléma motiválta: olyan vírusos megbetegedések felismerését, osztályozását tűztünk ki célul patológiai 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 rendelkeznek emberekhez, teljesen anonimizáltak, nem található a metaadatokban 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 megoldásunk kialakításánál (mivel egy páciensről mindig csak az egyik fajta nézőpont állt rendelkezésre), de a későbbi fejlesztések során lehetőségként tekintettünk rá. A többféle nézet kihasználását a 3. fejezet tárgyalja.

2.1 Adathalmaz

A teljes adathalmaz 958 darab felvételt tartalmaz összesen 19 betegségi osztályról. Adattisztítás (hibás, vízjeles stb. felvételek eltávolítása) után 15 osztályról 880 felvétel került a végső adathalmazba.

További speciális körülmény, hogy a különböző forrásokból származó képgyűjtemények eltérő felbontással rendelkeznek, a legkisebb mindössze 150x150, míg a legnagyobbak a 2500x2500 pixel nagyságot 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*, *chlamydomyces*, *covid*.

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

A Fig. 2 ábra felvételein jól felismerhetők a tüdő COVID-19 vírus okozta kóros elváltozásai. Bal oldalon: a tünetek felismerhetők a „sűrűbb” tüdő területekről, jobb oldalon pedig a „sűrűbb” területek hőtésképen vannak ábrázolva. A Fig. 3 ábrán pedig az egészséges, tüdőgyulladásos és COVID-19 tüneteket mutató felvételekből látható egy-egy mintakép.

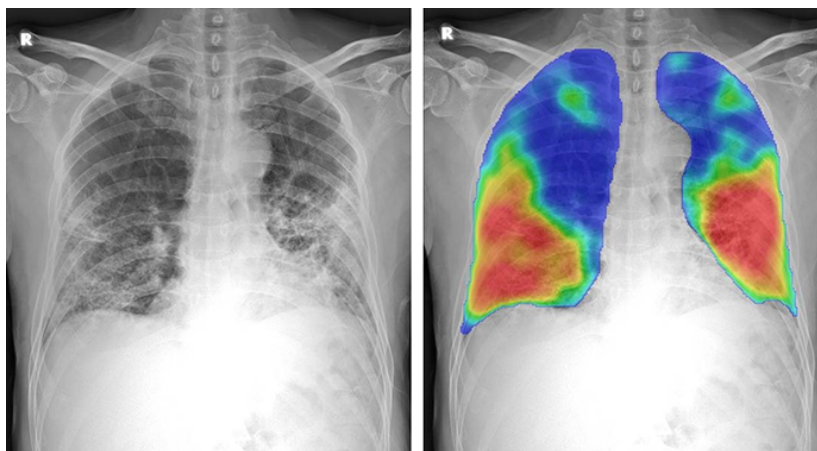


Fig. 2. COVID-19 tüneteket mutató mellkasröntgen felvétel (www.delft.care)

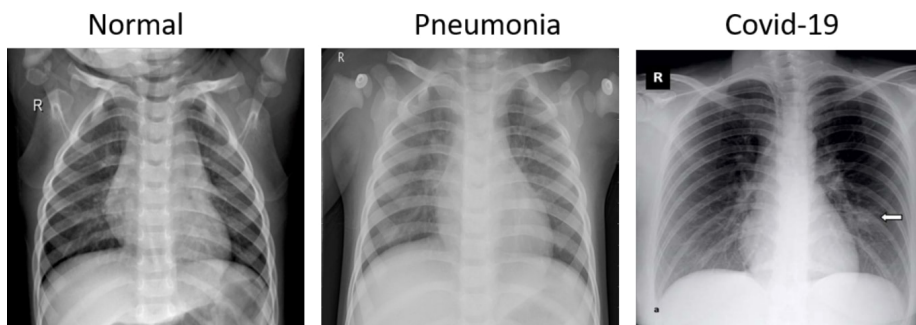


Fig. 3. Egészséges, tüdőgyulladásos és COVID-19 tüneteket mutató felvételek

2.2 Neurális 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 [14] publikáció mindösszesen annyit

osztott meg a jellemzővektorok leképezésére használt neurális architektúráról, hogy VGG [11] és Inception [13] hálózatokat használtak fel. Mi ettől a megközelítéstől eltértünk, mivel ezek általános objektum kategóriákra lettek megalkotva (és nem orvosi röntgen képekre). Így egy saját konvolúciós hálózatot alkalmaztunk, amely 8 rétegből épül fel (egy konvolúciós réteget mindig egy pooling réteg követ, és ezt 4-szer ismételtük egymás után), a 8 réteghez tartozó paraméterek száma 2300 volt a következő kernel méretekkel:

- konvolúciós réteg kernel mérete: 3x3 stride mérete: 1, padding: 1
- pooling réteg kernel mérete: 2x2, stride mérete: 2

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 előállítására érdekében. A két hálózat, f és g (tanító és teszt képek beágyazása) azonos architektúrával rendelkezik. Annak érdekében, hogy a CNN hálózat képes legyen az összes eltérő méretű képet feldolgozni, a minták egységesen 460 x 460 pixel nagyságra lettek középpontosan méretezve.

3 Double-View matching Network

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 [6][16]. Az új, két nézőpontot felhasználó megoldásunk (melyet *Double-View Matching Network*-nek, röviden *DVMN*-nek neveztünk 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 [4]. 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 [5], ugyanis a Matching Network több különálló vektorra való tanítás esetén a nézetek közötti különbségre tanulna rá a hasonlóságok helyett, így a megoldásunk a nézetek uniójával dolgozó alapötleten alapult. A következőkben megadjuk az egyes nézetek felhasználásának formális leírását.

3.1 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, az

előző fejezetben bemutatott, 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:

$$V_{L1} = \{v_{L1} | v_{L1} = f_{CNN1}(x), x \in S_{L1}\} \quad (3)$$

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:

$$V'_{L1} = \{v'_{L1} | v'_{L1} = f_{MN1}(v_{L1})\} \quad (4)$$

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:

$$V'_{U1} = \{v'_{U1} | v'_{U1} = f_{MN1}(f_{CNN1}(x)), x \in S_{U1}\} \quad (5)$$

Ha V'_{U1} elemeiből kiválasztjuk az osztálycímkével rendelkező vektorokat az ún. *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:

$$V_{L2} = \{v_{L2} | v_{L2} = f_{CNN2}(x), x \in S_{L2}\} \quad (6)$$

$$V'_{L2} = \{v'_{L2} | v'_{L2} = f_{MN2}(v_{L2})\} \quad (7)$$

$$V'_{U2} = \{v'_{U2} | v'_{U2} = f_{MN2}(f_{CNN2}(x)), x \in S_{U2}\} \quad (8)$$

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 $k \times 2m$ dimenziójú lesz.

Valós környezetet szimulálva az ideális összeállítás elvárása irreális követelmény lenne, ezért ennél a pontnál a következő lehetőségek álltak 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.

Saját módszerünket úgy alakítottuk ki, hogy ne legyen minimális mintaszám feltétel nézetenként, az egyetlen feltétel az, hogy $k_1 + k_2 \geq N$. Ez a megoldás a nézetek uniójának ötletével dolgozik (ezt a következő részben fejtjük ki) és kiküszöböli a hiányos minták problémáját, az adattábla dimenziója így $(k_1 + k_2) \times m$ lesz.

3.1 DVMN a többféle nézet kihasználásához

A DVMN módszer során az előző alfejezetben vázoltakhoz hasonlóan 2 darab CNN-t tanítottunk be, de az egyes vektorok összefűzése helyett a vektorok halmazainak uniójának ötletét használtuk fel. Legyenek a V_{L1} és a V_{L2} a (3) egyenlettel analóg módon a jellemzővektorok halmazai. A következőkben bemutatott megoldás a nézetek uniójának kihasználására építi fel a modellt. Vegyük a jellemzővektorok unióját:

$$V_L = V_{L1} \cup V_{L2} \quad (9)$$

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:

$$V'_L = \{v'_L \mid v'_L = f_{MN}(v_L), v_L \in V_L\} \quad (10)$$

Amennyiben egy ismeretlen képhalmazra szeretnénk a megtanult új vektorteret használni, akkor korábban megtanult CNN₁, CNN₂ (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:

$$V'_U = \left\{ v'_U \mid \begin{array}{l} v'_U = f_{MN}(f_{CNN1}(x)), x \in S_{U1} \\ v'_U = f_{MN}(f_{CNN2}(x)), x \in S_{U2} \end{array} \right\} \quad (11)$$

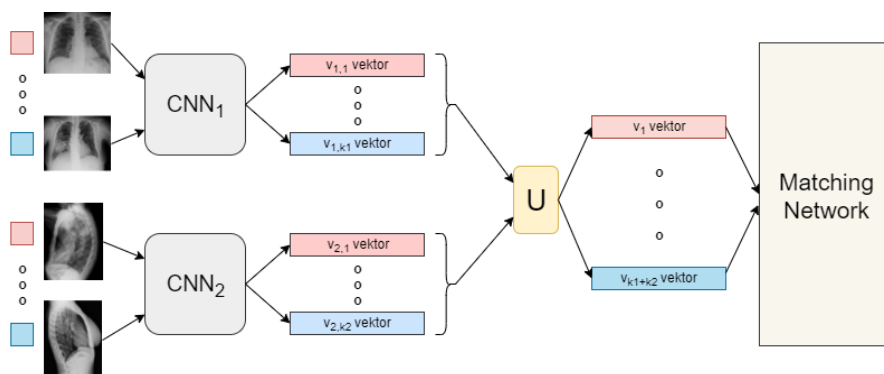


Fig. 4. Double-View Matching Network support vektor leképezése

A többféle nézetet hatékonyan kihasználó és a kiegyenlítetlen osztályok problémáját kezelni képes *Double-View Matching Network* architektúrájának *support* halmaz leképező része a Fig. 4. ábrán látható:

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.

4 Eredmények

A javasolt módszerünk (DVMN) és a módszerünkkel elkészült fejlesztések hatékonyságát tesztelendő, 3 különböző kísérleti környezetet terveztünk meg és az új architektúrát összevetettük az alapjaként szolgáló Matching Network (MN) modellel, illetve egy k -NN osztályozóval is. A beállítások a következők voltak:

- 2 külön nézet átlaga: az egyes nézetek felvételeit manuálisan elkülönítve készültek a mérések. Az egyes modellek bemeneteként csak egyfajta nézet képei kerültek kiválasztásra, majd a két-két modell eredményeit páronként átlagoltuk.
- Automatikus nézet szeparálással: az egyes nézetek felvételei automatikusan kerültek az alkalmazott modellek megfelelő alhálózatának bemenetére.
- Nincs nézet szétválasztás: az egyes nézetek felvételei szétválasztás nélkül kerültek a modellek bemenetére “vegyes” adathalmazokat alkotva ezáltal.

A tanító adathalmazok olyan módon kerültek összeállításra, hogy a felvételek egy számottevő részét alkotó COVID halmaz képei ne kerüljenek beválogatásra, ezáltal szimulálva a vírus, mint új betegség felismerési feladatát. Ezzel szemben a teszt halmazokba, azaz azon osztályok képei közé, amelyeket felismerni kívántunk szándékosan beválogattuk a COVID osztály képeit is. A következő *Táblázat 1*-ben láthatóak a mérési eredmények (accuracy pontossági értékek) a különböző tanító és teszt halmazok osztályainak összeállításait figyelembe véve. A teljes adathalmaz

számosságának korlátait kénytelen figyelembe véve 4, illetve 6 tanító osztályt és 4, illetve 2 darab teszt osztályt használtunk fel 1-2-5 darab minta beválogatásával osztályonként. A táblázatból látható, hogy a javasolt DVMN módszerünk 6 esetből 5-ször bizonyult a legjobb megoldásnak (továbbá átlagosan is) a hat alternatíva közül.

Táblázat 1. Különböző nézetbeállításokat használó implementációk összehasonlítása

<i>cellákban: accuracy értékek</i>	2 külön nézet átlaga		Automatikus nézet szeparálással		Nincs nézet szétválasztás	
tanító/teszt osztályok száma és shot száma	2 MN átlaga	2 k-NN átlaga	DVMN	Kétnézetes k-NN	MN	k-NN
C4/C4/S1	0,8373	0,5781	0,8672	0,5765	0,8240	0,6101
C4/C4/S2	0,8600	0,6197	0,8810	0,5534	0,8654	0,5500
C4/C4/S5	0,7971	0,6365	0,8141	0,6141	0,7970	0,6320
C6/C2/S1	0,7940	0,5433	0,8437	0,6500	0,8134	0,5967
C6/C2/S2	0,8571	0,6510	0,7952	0,5743	0,7701	0,6587
C6/C2/S5	0,8140	0,6000	0,8967	0,5123	0,8020	0,5363
Átlag	0,8266	0,6048	0,8497	0,5798	0,8120	0,5973

5 Összefoglalás

A cikkben egy aktuális gyakorlati problémán - új / ismeretlen betegségek mellkasröntgen felvételeinek elemzésén - keresztül mutattunk be egy új FSL technológiát, amely a Matching Network osztályozóra épült. A vizsgálatunk középpontjában az alap Matching Network architektúra teljesítményének felmérése és olyan irányú továbbfejlesztése állt, hogy az képes legyen különböző felbontású és nézetű mellkasröntgen felvételek hatékony osztályozására. A kutatásunk arra irányult, hogy hogyan lehet ismeretlen betegségeket osztályozni kizárólag más betegségek jellemzőinek felhasználásával mindössze néhány minta segítségével. Ennek legfontosabb kutatási eredménye a *Double-View Matching Network (DVMN)* neurális architektúra és osztályozó létrehozása volt, amely képes páciensfüggetlen, kétfajta nézőpontból készített felvételek hatékony jellemzőképezésére és rajtuk keresztül a betegségek felismerésére új metrikatanuló módszerek felhasználásával. Kísérleti eredményeinkben bemutattuk, hogy a javasolt DVMN módszer átlagosan a legjobb eredményt éri el a tesztelésben kipróbált 6 módszer közül, és 6 különböző tesztelési szcenárióban 5-ször ért el legjobb pontossági értéket.

Köszönetnyilvánítás

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 Együtműködések).

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] 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>
- [4] Geras, K. J., Wolfson, S., Shen, Y., Wu, N., Kim, S., Kim, E., ... & Cho, K. (2017). High-resolution breast cancer screening with multi-view deep convolutional neural networks. arXiv preprint arXiv:1703.07047.
- [5] Kan M, Shan S. and Chen X., (2016). Multi-view Deep Network for Cross-View Classification, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 4847-4855, doi: 10.1109/CVPR.2016.524.
- [6] Li, Y., Yang M. and Zhang Z., (2019). 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.
- [7] Mishra, N., Rohaninejad, M., Chen, X., & Abbeel, P. (2017). Meta-Learning with Temporal Convolutions. ArXiv, abs/1707.03141. <https://arxiv.org/abs/1707.03141>
- [8] Parag: Metric learning tutorial https://parajain.github.io/metric_learning_tutorial
- [9] 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>
- [10] Shyam, P., Gupta, S., & Dukkipati, A. (2017). Attentive Recurrent Comparators. ArXiv, abs/1703.00767. <https://arxiv.org/abs/1703.00767>
- [11] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556. <https://arxiv.org/abs/1409.1556>
- [12] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical Networks for Few-shot Learning. ArXiv, abs/1703.05175. <https://arxiv.org/abs/1703.05175>
- [13] 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>
- [14] Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. arXiv preprint arXiv:1606.04080.
- [15] Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys (CSUR), 53(3), 1-34.
- [16] 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/CVF Conference on Computer Vision and Pattern Recognition, pp. 8543-8553.