
Tüdőgyulladás eredetének megállapítása mellkas röntgen képek alapján

Kis Máté Levente
Budapesti Műszaki és
Gazdaságtudományi
Egyetem
mkis@edu.bme.hu

Réfi Nándor
Budapesti Műszaki és
Gazdaságtudományi
Egyetem
refinandor@edu.bme.hu

Szócska Miklós Bendegúz
Budapesti Műszaki és
Gazdaságtudományi
Egyetem
miklosbendeguzszocska@edu.bme.hu

Összefoglaló

A 2019. december 31-én megjelent COVID-19 vírus világszerte nagy kihívások elé állítja az országok egészségügyi szerveit és annak alkalmazottait. Nő az igény a betegek gyors és pontos diagnosztizálására, valamint fontos a vírus okozta tünetek kiismerése. A betegség egyik gyakori tünete a tüdőgyulladás, amelyet nem csak vírusok, de gyakran baktériumok vagy gombák is okozhatnak. A különböző típusú tüdőgyulladások felismerése kihívást jelenthet vagy időigényes lehet. Úgy gondoljuk érdemes olyan Deep Learning modellek kifejlesztésére törekedni, amelyek képesek azonosítani a tüdőgyulladás eredetét egy mellkas röntgen alapján. Az általunk létrehozott modellel három kategóriába eső képeket vizsgálunk: COVID-19 eredetű tüdőgyuladást, bakteriális eredetű tüdőgyuladást és tüdőgyulladás mentes röntgen képeket. A modellhez tartozó forráskód az alábbi linken érhető el:

https://github.com/SonOf1998/COVID-19_detection_in_lungs

1 Hasonló munkák

Az alábbi linken elérhető egy projekt cikke, amelyhez hasonlót terveztünk mi is készíteni:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7187882/>

A készítő a miénkhez hasonló adatbázison akartak bináris (COVID, No findings) illetve multi-class (COVID, No findings, Pneumonia) klasszifikációt végezni. A képek mintázatainak megtalálásra ők saját konvolúciós hálózatot fejlesztettek a Darknet-19 továbbfejlesztésével, amellyel heatmapeket is képesek voltak csinálni, hogy annak segítségével egy specialista (radiológus) is meggyőződhesen a klasszifikáció megbízhatóságáról.

Az említett munka a bináris klasszifikáció esetében 98.05%-os, multi-class klasszifikációkor pedig 87.02%-os pontosságot tudott felmutatni.

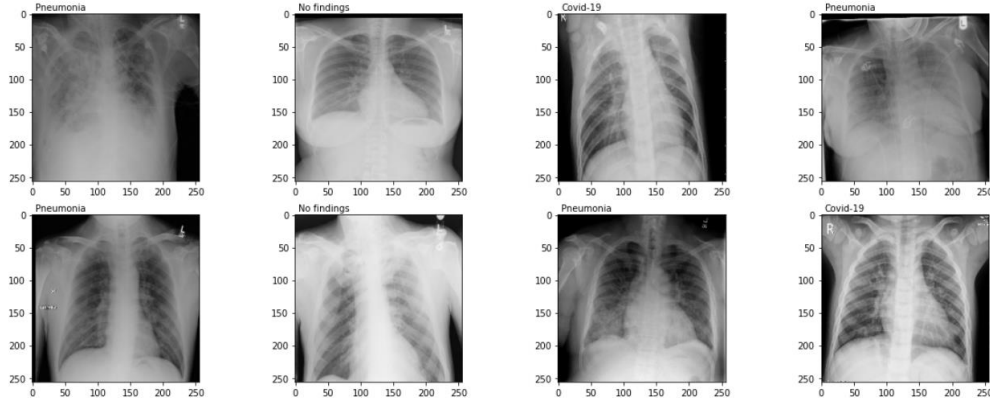
2 Adatelőkészítés

Az adatbázisunkat úgy szerettük volna kialakítani, hogy az közel azonos mennyiségű szemből nézetes, egyenes állású mellkasi röntgent tartalmazzon mind a három kategóriára vonatkoztatva. A képeket a repository *image_datasets* mappájába helyeztük, amely a repository zipként letöltésével Google Colab platformra is egyszerűen átvihető.

2.1 Felhasznált adatbázisok

A tanításhoz és kiértékeléshez felhasznált adathalmazunk 4 különböző publikusan elérhető adatbázisból lett összeillesztve, amelyeknek a linkje elérhető a github repository leírásában. Úgy tapasztaltuk, hogy ezek az adatbázisok kis számú duplikátumot is tartalmaznak, így ezeket még előkészületként egy scripttel kiszűrtük és eltávolítottuk. Némely kép orientációja jócskán eltért a többitől, ezeket megpróbáltuk visszaforgatással fixálni, amelyiknél azonban ez nem volt lehetséges, azt eltávolítottuk.

Az adatbázisunk végeredményben közel 450 COVID-19 vírusfertőzött, 500 tüdőgyulladásos illetve 500 egészséges tüdő röntgenképét tartalmazza.



2.2 Training-Validation-Test felosztás

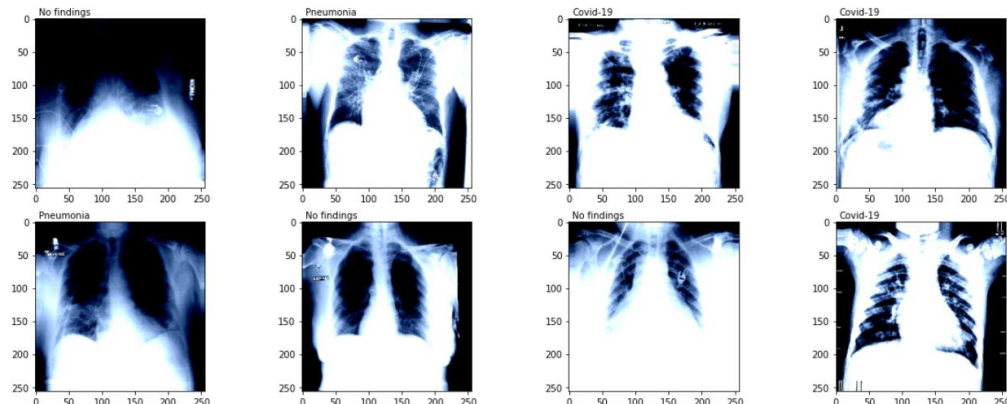
Az adatbázisról a test set leválasztását még előzetesen végeztük el, a training és validation szétválasztása pedig 92% és 8%-os arányban történt a training set javára.

Össességében 1169 képet tartalmaz a training set, 100-at a validation set és 137-et a test set.

2.2 Augmentálás

Az augmentáció paramétereinek megadása előtt alkalmaztuk először preprocesszáló funkcióként a célmodell (SE-ResNet152) dedikált input transzformáló funkcióját.

Az augmentálás maga egy minimális forgatást (3°) és belenagyítást (10%) tartalmaz kiegészülve egy nagyobb változtatás megengedésével a fényerősség kapcsán, ami az eredeti intenzitás 85%-ától 120%-áig terjedhet.



3 Probléma megközelítése

A probléma megoldására nem volt kérdés, hogy transfer learninget szeretnénk alkalmazni. Próbaképpen az ImageNet Large Scale Visual Recognition Challenge (ILSVRC) résztvevő modelljeit használtuk a súlyok átvitelével. Ezek meglepően jól teljesítettek, így ezen modellek között válogattunk.

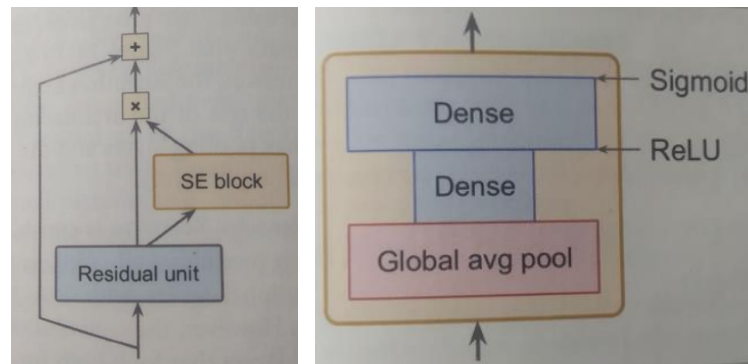
3.1 Eddig kipróbált modellek

A kipróbált modelljeink mérete széles skálán ugrál, viszont közös jellemzőjük, hogy az ImageNet súlyokkal inicializáltuk mindegyiket. A kipróbált modellek a VGG19, DenseNet201, InceptionV3, NASNet-Large, ResNet34, ResNet152 v2 illetve a SE-ResNet152 voltak.

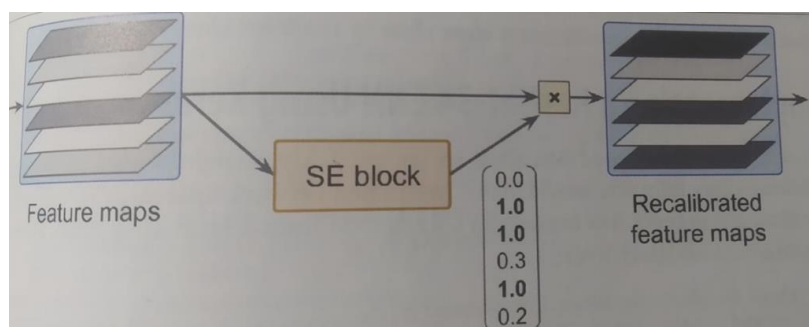
3.2 SE-ResNet152 hálózat

Amit a legideálisabbnak találtunk a végén mind performanciát, mind tanítási időt illetően az a SE-ResNet152 volt.

A 2017-es ImageNet challenge győztes architektúrája a Squeeze-and-Excitation Network (SENet) volt, amely eddig meglévő architektúrák teljesítményét volt képes nagymértékben növelni. Az SENet a meglévő modellek minden unitjához egy ún. SE blockot ad, amely magában is egy mini neurális háló és felépítése a lenti ábrákon látható.



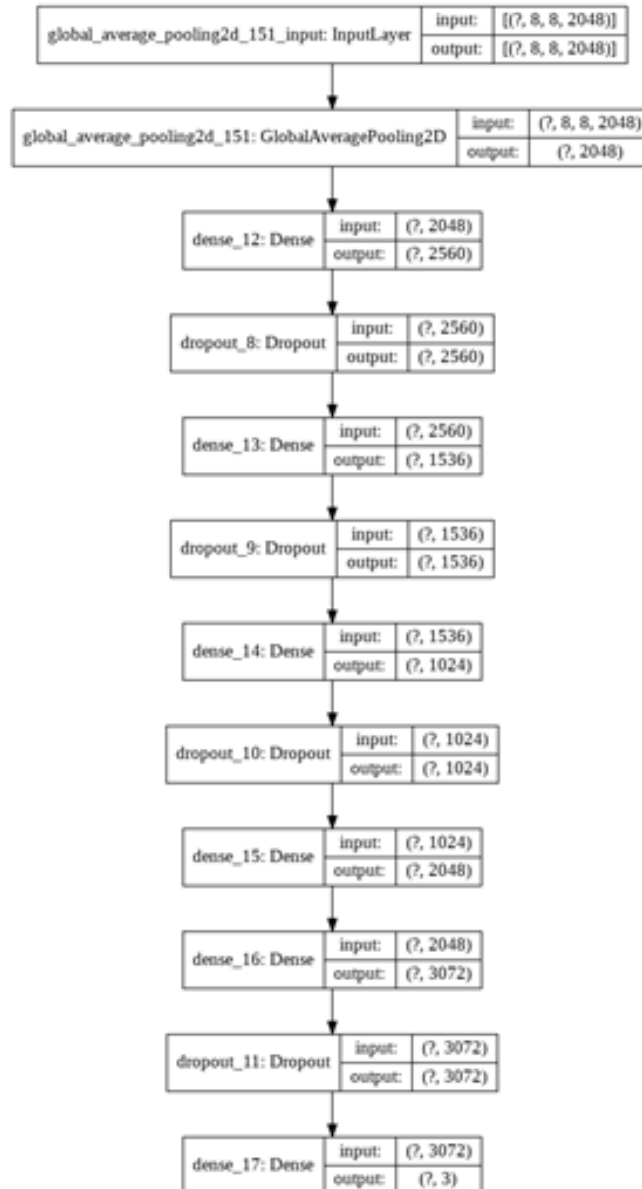
A SE block hatása, hogy a feature mapeket (mélység dimenziót) áttekintve megtanulja, hogy melyik feature-ök voltak egyszerre aktívak és ennek megfelelően fogja erősíteni a releváns mapeket, illetve gyengíteni az irrelevánsakat (rekalibráció).^[1]



3.3 Felső Dense rétegek

Az SE-ResNet152 tetejére egy 3 kimenetű neurális hálót tettünk, amely némely rétegében dropout regularizációt végez. A hidden layerok közös jellemzője, hogy aktivációjuk Leaky ReLU, míg utolsó réteg softmax aktivációt használ.

A dense rétegek unit számát, illetve a dropout rátákat hiperparaméter optimalizálás útján határoztuk meg. (4.2)



4 Tanítás és kiértékelés

Több tanítást is lefuttattunk és kísérletezéssel próbáltuk elérni a lehető legjobb eredményeket. Több optimalizációs algoritmust, aktivációt is kipróbáltunk. Változtattuk a neuronok számát és dropout rétegeket vezetünk be, valamint hiperparaméterek optimalizálására szánt könyvtárakat kihasználva finomítottuk tovább a modellünket.

4.1 Tanítás menete

A tanítások során azt véltük felfedezni, hogy a legjobb eredményeket úgy érjük el, ha SE-ResNet152 rétegeit befagyasztjuk és csak a top dense réteget tanítjuk tovább.

A modellhez kipróbáltuk mind az Adam^[2], SGD^[3] és AdaBound^[4] optimalizációs algoritmusokat, melyek közül az AdaBound bizonyult a leggyorsabb és legnagyobb accuracy-t biztosító algoritmusnak.

Checkpointing módszert használtunk, hogy tanítás végén a modell legjobb állapotát értékeljük ki, valamint Early Stopping-ot, hogy overfitting esetén leállítsuk a tanítást. A tanítás folyamatát logoljuk, amelyet később Tensorboardon jelenítettünk meg.

4.2 Hiperparaméter optimalizálás

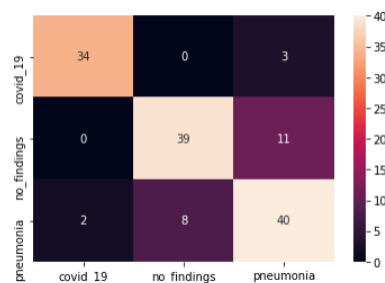
A top dense rétegek neuronszámát és a dense rétegek közötti dropout rétegek rate-jét optimalizáltuk a Keras-Tuner random search algoritmusát alkalmazva. Az algoritmust 20 iterációig futtattuk, amelyek közül a legjobb validation accuracy-val bíró hiperparaméterek együttesét választottuk.

5 Eredmények vizualizálása

Az eredmények vizuális ábrázolása sokat segíthet a modell javításában és az eredmények értelmezésében, így mi is több technikát alkalmaztunk ilyen célokra.

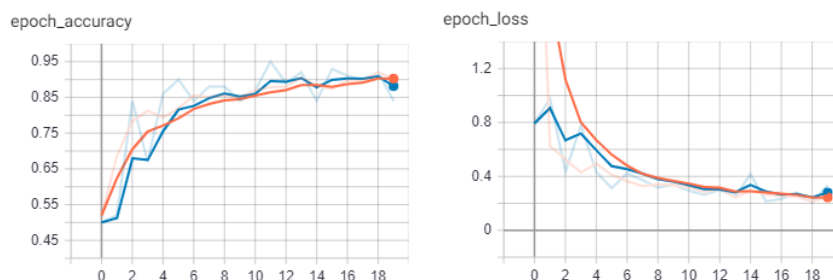
5.1 Konfúziós mátrix

Az egyik legismertebb technika az úgynevezett konfúziós mátrix, ebből azt olvashatjuk le, hogy melyik osztályokat milyen osztálynak tippelte a modell, és így következtethetünk rá melyik képekkel van problémája a modellnek és esetleg hol tudunk javítani rajta.



Egyértelműen látszik, hogy míg a covid-os képeket közel tökéletesen felismeri, a pneumóniás és egészséges képek felismerésével nehézségei vannak.

5.2 TensorBoard - Tanítás és tesztelés

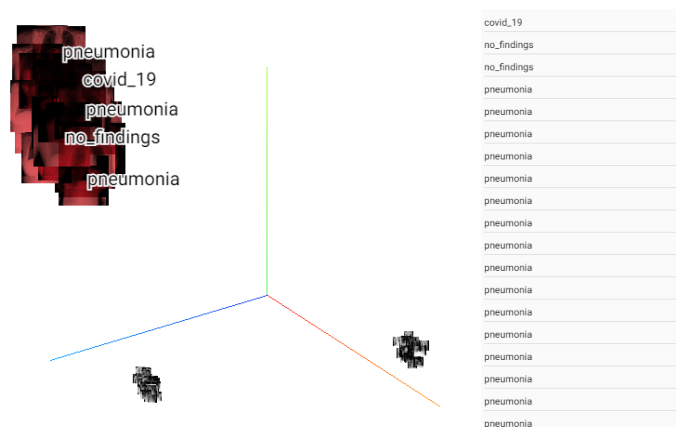


Az ábrákon a tanítás alatt epoch-onként logolt pontosság és hiba értéke látható. A narancssárga vonalak a tanítás eredményei, a kékek a validációé. Látszik, hogy a modell gyorsan tanul és beáll egy optimális értékre, azonban ennél még lehetnének jobb eredményei is további fejlesztések árán. A vége felé, a 18 epoch környékén, túltanulás jelei is megfigyelhetők, de ez ellen védett, hogy a legjobb validációs pontosságot produkáló modell súlyait mindig mentettük.

A betanított modell a validációs dataset-en 95%-os, míg a teszt seten 82.5%-os pontosságot ért el.

5.3 TensorBoard – Projection

Használtunk projekciót is az utolsó konvolúciós réteg kimenetelén. Lényegében predikáltunk a teszt képeken, és az utolsó konvolúciós réteg kimenetelét embedding-é alakítva, azokon futtatunk egy dimenzió csökkentő algoritmust, hogy így 3D-ben tudjuk megjeleníteni. Azt vizsgáltuk, hogy mennyire alkotnak csoportokat a kimenetek, és azok mennyire különböznek el egymástól.



A két képen egy UMAP algoritmust használó futás eredménye látható. Jól elkülöníthető csoportok képződnek, és a kijelölt csoportban található képek osztályait vizsgálva, (amelyek a jobb oldali képen láthatóak,) azt látjuk, hogy nagyrészt azonos osztályba tartozó képek kerültek egy csoportba, így a modellünk jól elkülöníti őket.

5.4 GradCAM

Egy olyan módszer, ami azon alapszik, hogy az utolsó konvolúciós réteg aktivációit vizsgáljuk és deriválással visszafejtjük. Lényegében egy olyan hőtérképet nyerhetünk ki képekhez, amik azt mutatják meg, az egyes tartományai mennyiben járultak hozzá a modell legnagyobb valószínűségű predikciójához. Leegyszerűsítve azt vizsgáljuk, miért azt tippelte a modell, amit, így ellenőrizve, hogy a megfelelő problémát oldja-e meg.



Két kép látható, melyekre helyes predikció született. Az alsó sarokban látható a (helyes) osztály és a hozzá tartozó valószínűség. Arra tudunk következtetni a képekből, hogy a modell valóban a tüdőben fellelhető mintázatokat vizsgálja és az alapján dönt, ami siker. Azonban látható, hogy a váll környékét is vizsgálja, ahol felirat látható sok képen, így időnként hibázik is szemantikai értelemben, mert a feliratnak nincs köze a betegségekhez. (Ha mégis van köze a feliratnak a betegséghez egyes esetekben, az még rosszabb lehet, mert olyan információt használ, amit nem lenne szabad.)

6 Konklúzió

A projektünk során pácienseket a mellkasi röntgengképeik alapján soroltuk be három kategóriába megbízható, de még fejlesztendő 82.5%-os pontossággal a test seten. A modell teljesítménye az adatbázisunk (és így a training set) növelésével arányosan javulna, így érdemes szemmel tartani a népszerű dataset oldalakat (Kaggle, AWS Open Data, stb.) illetve további röntgengképek után kutatni kevésbé ismert forrásokon is.

Referenciák

- [1] Aurélien Geron (2019) Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow 2nd edition
- [2] Diederik P. Kingma & Jimmy Lei Ba. Adam (2015) A method for stochastic optimization, *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*
- [3] Herbert Robbins & Sutton Monro (1951) A stochastic approximation method, *The Annals of Mathematical Statistics*, 22(3):400-407
- [4] Liangchen Luo, Yuanhao Xiong, Yan Liu & Xu Sun (2019) Adaptive Gradient Methods with Dynamic Bound of Learning Rate, *International Conference on Learning Representations (ICLR) 2019 Conference*