

Mélytanulás házi feladat dokumentáció

BMEVITMMA19-HU

1. Bevezetés

Az élelmiszeriparban számos gyakorlati felhasználási területen kínál előnyt a képfelismerő alapú alkalmazások egyik fontos kihívása, az étel képek automatikus osztályozásának módszere. Példaként említhető az éttermekben történő gyorsabb és pontosabb rendelésfeldolgozás, a táplálkozási tanácsadók és egészségügyi alkalmazások fejlesztése, valamint az e-kereskedelem hatékonyságának növelése szempontjából kiemelten hasznos lehet egy pontos kép osztályozó rendszer.

Az alább bemutatott projektben egy mélytanulás-alapú megoldást dolgoztunk ki, amely három különböző ételt képes felismerni: a hamburger, bagel, vagy szendvics kategóriákat.

Célunk egy olyan modell kifejlesztése volt, amely kellő pontossággal képes különböző ételeket osztályozni, ám eközben hatékonyan használja az erőforrásokat is, illetve adaptálható különböző környezetekben való használatra. Ennek érdekében a fejlesztési folyamat során különféle neurális hálózati architektúrákat vizsgáltunk és optimalizáltunk.

1.1 Kapcsolódó munkák

A munkánk során alapvető referenciaként használtuk a [Deep Residual Learning for Image Recognition](#) [1] című tanulmányt, amely a ResNet-ek mély tanulási architektúrájának hatékonyságát és optimalizálhatóságát mutatta be. Ez a tanulmány nyújtott inspirációt az általunk alkalmazott modell kialakításához is.

A tanulmány alapján megállapítható, hogy a mélyebb hálók használata jelentős javulást eredményezhet a vizuális feladatok, például az osztályozás és a detektálás pontosságában. Mivel az osztályozási feladatunk célja az ételek képeinek azonosítása, így az olyan mélyebb architektúrák, mint a ResNet18, lehetővé teszik a modell számára, hogy komplexebb jellemzőket tanuljon meg, ami jelentősen növeli a pontosságot.

A tanulmány továbbá rámutat arra is, hogy az ensemble megközelítések, amelyek több háló kombinációját alkalmazzák, további teljesítmény javulást képesek elérni, tehát az ensemble technikák használata előnyös lehet nagyobb kategória számú problémák esetén is, amely a munkánk szempontjából különösen releváns megállapítás. A tanulmány azt is hangsúlyozza, hogy a ResNet-ek bár mélyebbek, de alacsonyabb számítási komplexitással is rendelkeznek, összevetve például a VGG hálókkal.

Továbbá jelentős szerepet játszott még a projekt tervezésében a PyTorch Lightning CIFAR-10 Baseline példája [2] is, amely részletekbe menően mutatja be a PyTorch Lightning használatának előnyeit a mélytanulási modellek fejlesztésében és strukturált kezelésében. A tananyag szemlélteti a keretrendszer funkcióit (például a DataModule és Trainer osztályok

használatát), valamint a modellépítés és optimalizáció egyszerű, mégis hatékony megközelítését is kifejti.

A források együttes feldolgozása nem csupán elméleti alapul, de gyakorlati útmutatóul is szolgált a saját osztályozási rendszerünk kialakításához, amely során a kapcsolódó munkákból tanultakat a feladatunk sajátosságaira szabott implementációban használtunk fel.

2. Tanítás

A feladat során négy különböző megközelítést alkalmaztunk a modellek tervezésére, mindegyik más szempontból próbálta maximalizálni a pontosságot és tanulási hatékonyságot.

2.1 Baseline model

Az első modellünk egy alap konvolúciós neurális háló (CNN) volt, amelyet a baseline teljesítmény gyors felmérésére terveztünk, egyszerű architektúrája az alapvető eredmények szolgáltatását célozta, különösebben bonyolult technikák nélkül.

2.2 Komplex model

A második modell egy többrétegű CNN architektúra volt, amely már haladó technikákat, például dropout rétegeket is tartalmazott. Az itt hozzáadott rétegek csökkentették a túltanulás mértékét, amely a kis tanítóhalmaz miatt kritikus tényező volt. A többrétegű felépítés által a modell bonyolultabb mintázatokat is felismert, míg a dropout rétegek biztosították a tanulási folyamat általánosítási képességét.

2.3 Transfer learning model

A harmadik megközelítésünk a transfer learning technikán alapult, amely során előre betanított ResNet18 modellt finomhangoltunk a feladathoz. Ez az architektúra, amely a mélytanulás egyik legszélesebb körben alkalmazott modellje, a már megtanult alapvető képi jellemzők kihasználásával rövidebb idő alatt jóval magasabb pontosságot ért el, így az alkalmazásával gyors és hatékony tanulást valósíthattunk meg.

2.4 Ensemble model

Az Ensemble modellek célja, hogy a különböző modelleket kombinálva javítsuk a teljesítményt és csökkentsük az egyes modellek hibáinak hatását. A házi feladat során az előbbiekben említett három különböző modellt használtuk és ezek eredményét egyesítettük egy ensemble megoldásban a legjobb teljesítmény elérése érdekében. A modell kimenetének előállításakor minden egyes alapmodell kimenetei átmenetileg összegzésre kerültek, majd az összes kimenet átlagát vettük. Így egy végső konszenzusos eredményt kaptunk, ami miatt az ensemble modellünk stabilabb és robosztusabb lett a hibákat tekintve.

2.5 A tanítás folyamata

A tanítás folyamata során a PyTorch Lightning keretrendszert hívtuk segítségül a modellek implementációjához illetve karbantartásához. A képeket a tanítás előtt egységesen 224x224 pixel méretűre skáláztuk, és normalizáltuk, hogy a bemeneti értékek konzisztens tartományba kerüljenek a modell hatékony tanulása érdekében. A tanító és validációs adathalmazt 80-20%-os arányban osztottuk fel, osztályonként 80 tanító- és 20 validációs mintával.

Ezután a tanítás során keresztentrópia loss függvényt alkalmaztunk, amely az osztályozási problémák esetén ideális választás, hiszen minimalizálja az eltérést a modell által becsült és a valós osztályok valószínűségei között, növelve ezzel a predikciók pontosságát. Az optimalizálásra az Adam algoritmust használtuk, amely az egyik legnépszerűbb módszer a mélytanulásban. A tanulási ráta $1e-3$ -ra, a súlycsökkenés pedig $1e-5$ -re lett állítva, hogy elősegítsük a stabil konvergenciát és elkerüljük a túlilleszkedést.

Az így kialakított modelleket vizsgáltuk a kiértékelési folyamat során.

2.6 Hiperparaméter optimalizálás

A modell teljesítményének maximalizálása érdekében részletes hiperparaméter-optimalizálást végeztünk a paraméterek monitorozásával és vizualizálásával a Weights and Biases (WandB) keretrendszert használva. Az optimalizációs folyamat célja a validációs teljesítmény javítása volt, tekintettel a modell tanulási hatékonyságára és általánosítási képességére, míg a hiperparaméter-söprések (sweep) során különböző konfigurációkat próbáltunk ki, hogy megtaláljuk az optimális beállításokat a tanulási ráta, a dropout arány és a batch méret számára.

A tanulási ráta esetében a megfelelő érték kiválasztása fontos feladat volt, hiszen meghatározta a modell súlyainak frissítési sebességét az egyes iterációk során. Az alacsony tanulási ráta stabil, de lassú tanulást eredményez, míg a túl magas érték az optimalizáció instabilitásához és oszcillációhoz vezet, így különböző értékekkel kellett tesztelnünk, és a kapott eredmények alapján választottuk ki az optimális konfigurációt.

A dropout arány szintén kulcsszerepet játszott az overfitting csökkentésében. Ez a regularizációs technika a neurális hálózat véletlenszerű neuron-kikapcsolását alkalmazza a tanítás során, segítve a modell általánosító képességét. A vizsgálatok azt állapítottuk meg, hogy a túl alacsony dropout arány nem nyújt elegendő védelmet a túlilleszkedés ellen, de a túl magas érték a tanulási képesség csökkenéséhez vezethet.

A batch méret hatását is kielemeztük: a nagyobb batch méretek előnye a gyorsabb tanulás és a simább gradiens frissítés volt, míg a kisebb batch méretek változatosabb frissítéseket tettek lehetővé, ami pozitív hatással volt a modell robusztusságára. Az optimalizáció során az egyes batch méret konfigurációk validációs teljesítményre gyakorolt hatását elemeztük, majd ez alapján állapítottuk meg az adott architektúrák számára legmegfelelőbb beállításokat.

Végül a hiperparaméter-optimalizálás során a WandB sweep funkciója lehetővé tette, hogy a különböző konfigurációkat gyorsan és hatékonyan kezeljük, miközben részletes adatokat is nyertünk általa a modellek teljesítményéről, így az optimalizáció

eredményeképpen sikerült olyan hiperparaméter-beállításokat azonosítanunk, amelyek a legjobb validációs pontosságot eredményezték.

Deployment környezetben a sweep folyamatok kikapcsolásra kerültek erőforráskorlátozási célból, és fix, statikus paraméterek alkalmazásával futtattuk a modelleket. Így a tanítási és kiértékelési folyamat alacsonyabb számítási igénnyel is működik, miközben a korábban optimalizált beállítások továbbra is garantálják a modell megbízhatóságát és teljesítményét.

3. Kiértékelés

Az elkészült modellek teljesítményét egy részletes kiértékelési folyamat segítségével elemeztük, amely különböző osztályozási mutatókat (pontosság, precízió, visszahívás, F1 mutató) és vizualizációkat tartalmaz. A kiértékelés során célunk az volt, hogy ne csak a modellek általános teljesítményét mérjük, hanem az esetleges gyenge pontokat is azonosítani tudjuk, továbbá a teszt adathalmazon végzett értékelés során az osztályok közötti helyes és helytelen predikciókat egy tévesztési mátrix segítségével ábrázoltuk, amelyet hő térkép formájában vizualizáltunk, így képesek voltunk azonosítani az osztályok közötti hibákat illetve felkutatni és lehetőség szerint javítani a hibákat okozó kódrészeket is.

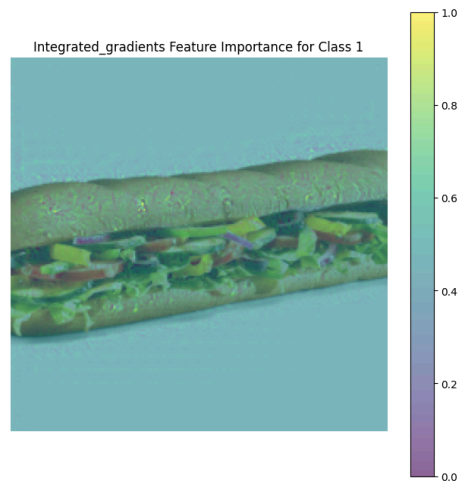
Az osztályozási riport, amely az egyes osztályok pontosságát, precízióját, visszahívását és F1 mutatóját tartalmazza, különösen hasznos volt az alulteljesítő osztályok azonosítására, így például a modell hajlamos volt összekeverni a hasonló vizuális jellemzőkkel rendelkező osztályokat, mint a bagel és a szendvics, amelyet a tévesztési mátrix és az osztályozási riport egyaránt megerősített.

A kiértékelési folyamat során vizualizáltuk a hibás predikciókat is, amely lehetővé tette a modell teljesítményének még mélyebb megértését, hiszen a rosszul kategorizált képeket elemezve kiderült, hogy az árnyékolás, a különböző hátterek, valamint az egyes osztályok vizuális átfedése befolyásolja leginkább a predikció pontosságát.

A vizualizációkat oszlopdiagramokkal is kiegészítettük, amelyek az egyes modellek (Baseline, Complex, Transfer Learning, Ensemble) teljesítményét mutatták be a pontosság, precízió, visszahívás és F1 mutatók alapján. Ezek az ábrák jól szemléltették a modellek közötti különbségeket, és segítettek az Ensemble modell előnyeinek kiemelésében, amely a különböző architektúrák erősségeinek kombinálásával érte el a legjobb eredményeket.

Az értékelési folyamatot az `evaluate_model` függvénnyel valósítottuk meg, amely az eredmények vizualizálására is képes, beleértve a tévesztési mátrixot és a rossz predikciók elemzését, míg a `compare_models` függvény segítségével az egyes modellek teljesítményét is összehasonlítottuk. Az Ensemble modell teljesítménye kiemelkedő volt, azonban a Transfer Learning modell bizonyult a legjobbnak, 90% feletti pontosságot és kiváló F1 mutatót érve el a teszt adathalmazon.

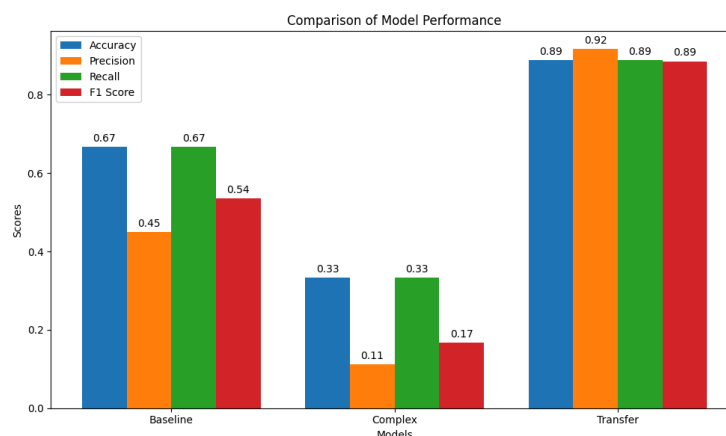
A vizualizációk között kiemelkedő volt az aktivációk és a jellemzők fontosságának megjelenítése, amelyet az *integrated gradients* módszerrel valósítottunk meg. Ez lehetővé tette, hogy láthatóvá tegyük, mely képrészletekre fókuszált a modell az egyes döntések meghozatalakor. Például az alábbi ábra egy szendvics kategóriába tartozó kép elemzését mutatja, ahol a modell a kenyér és a töltelék részleteire összpontosított, megerősítve, hogy ezek a vizuális jellemzők kritikusak a helyes predikcióhoz:



A kiértékelési folyamat során több eszközt is alkalmaztunk az eredmények elemzésére és vizualizálására. Az osztályok közötti hibák feltárásához tévesztési mátrixot használtunk, amely grafikus formában mutatta be a helyes és helytelen predikciók eloszlását, segítve az alulteljesítő osztályok azonosítását. Az egyes osztályok teljesítményének részletes elemzéséhez osztályozási riportot generáltunk, amely tartalmazta az osztályok pontosságát, precízióját, visszahívását és F1 mutatóját.

Classification Report:				
	precision	recall	f1-score	support
bagel	1.00	0.67	0.80	3
burger	1.00	1.00	1.00	3
sandwich	0.75	1.00	0.86	3
accuracy			0.89	9
macro avg	0.92	0.89	0.89	9
weighted avg	0.92	0.89	0.89	9

A modellek közötti teljesítménykülönbségek átlátható bemutatásához pedig oszlopdiagramokat készítettünk, amelyek vizuálisan szemléltették a mutatók közötti eltéréseket.



Emellett a rossz predikciókat külön is elemeztük, hogy azonosítsuk azokat a jellemzőket, amelyek nehézséget okoztak a modellek számára.

True: bagel, Predicted: sandwich

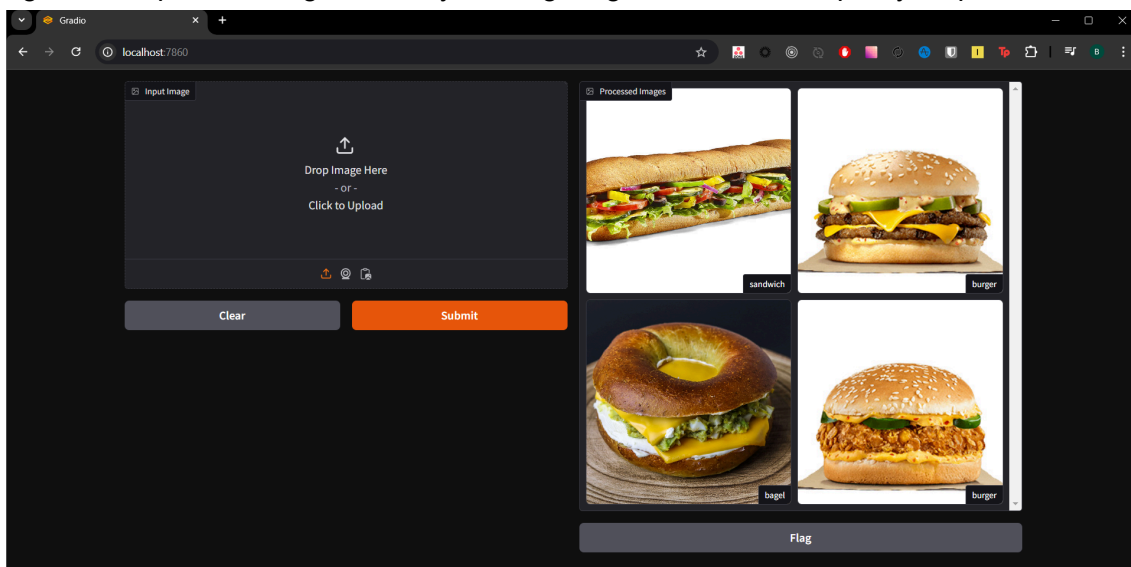


Deployment környezetben a vizualizációk helyileg kerültek mentésre, így az elemzések kevesebb erőforrással rendelkező rendszereken is futtathatók voltak.

A kiértékelési folyamat során felderítettük a modellek erősségeit és gyengeségeit, és ezek alapján meghatározhattuk a további fejlesztési lehetőségeket is.

4. Szolgáltatás létrehozása

A legjobb teljesítményű modellből egy felhasználói felülettel rendelkező webes szolgáltatást építettünk a gradio könyvtár segítségével. A felület képernyőképe:



A felhasználóknak lehetősége van képeket feltölteni, amelyekről a “Submit” gomb megnyomására, a feladatunkban megvalósított modell eldönti, hogy a három osztályból melyikbe tartoznak. Ezt követően a már feltöltött képek egy galériában jelennek meg a jobb oldalon, ahol a prediktált címke is szerepel a képeken. A “Flag” gomb segítségével pedig lehetősége van a felhasználónak a predikciókat egy csv fájlban elmenteni:

Input Image	Processed Images	timestamp
	[[{"image": ".gradio/flagged/Processed Images/95c9cfc97c93f416c0ae/test_4.jpg", "caption": "sandwich"}, {"image": ".gradio/flagged/Processed Images/cd609c5b00f2d8655dce/test_6.jpg", "caption": "burger"}, {"image": ".gradio/flagged/Processed Images/01e3098c19c7e610b634/test_8.jpg", "caption": "bagel"}, {"image": ".gradio/flagged/Processed Images/8cd78b06b72b84a7b088/test_5.jpg", "caption": "burger"}]]	2024.12.08 12:13

5. Konklúzió

A projekt eredményeképpen egy erőteljes és pontos ételképosztályozó rendszert fejlesztettünk. Az átviteli tanuláson alapuló modell bizonyult a legjobbnak, amely 90% feletti pontosságot ért el a teszt adathalmazon. Az Ensemble modellek tovább javíthatják az eredményeket, ha a különböző modellek erősségeit egyesítjük. A hiperparaméter optimalizálás és a vizualizációs technikák kulcsfontosságúak voltak a modellek finomhangolásában és értékelésében. Végül a kialakított modellből egy webes szolgáltatást hoztunk létre amelyen keresztül egy egyszerű, felhasználóbarát felületen lehet képeket osztályozni a modell segítségével.

A megoldás során a forráskód kommentelésére használtunk LLM-et (Github Copilot).

6. Hivatkozások

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). [1512.03385] *Deep*

Residual Learning for Image Recognition. arXiv. Retrieved November 30, 2024, from

<https://arxiv.org/abs/1512.03385>

[2] *PyTorch Lightning CIFAR10 ~94% Baseline Tutorial PyTorch Lightning*

Documentation. Retrieved December 01, 2024, from

https://lightning.ai/docs/pytorch/stable/notebooks/lightning_examples/cifar10-baseline.html