Készítette: Bata Dániel & Bunda Boldizsár
2024.12.06.

Image classification "in-the-wild"

1. Bevezetés

A projekt célja egy képosztályozási modell létrehozása, amely képes megkülönböztetni három különböző kategóriát: cipők, bútorok és könyvek. Az osztályozás valós környezetben készült képek alapján történik, amelyeket manuálisan gyűjtöttünk össze és osztottunk szét. Az alapötlet az, hogy egy egyszerű, de hatékony megoldást nyújtsunk egy valós képadatokkal végzett osztályozási feladatra. A tudományos hátteret a mély tanulási modellek képfeldolgozási alkalmazásai, mint például a MobileNetV2 architektúra, valamint az adat augmentáció hatása a modellek teljesítményére adják meg. A projekt során modern keretrendszerek, mint például a PyTorch Lightning és a Weights & Biases segítették a modell fejlesztésében és értékelésében.

2. Módszertan

2.1. Adatforrás és előfeldolgozás

A projekt egyik legfontosabb eleme az adatforrás és az adatok előfeldolgozása, mivel a mély tanulási modellek hatékonysága és általánosító képessége nagyban függ az alkalmazott adatok minőségétől és előkészítésétől. Feladatunk során különös figyelmet fordítottunk arra, hogy az előfeldolgozási lépésekkel optimalizáljuk a modell tanulását.

2.1.1. Adatforrás

Az adathalmaz a projekt keretében sajátkezűleg készült, valós képekből áll, amelyek három kategóriába sorolható objektumokat tartalmaznak:

- **Cipők:** különféle típusú és stílusú lábbelik.
- **Bútorok:** például asztalok, székek vagy egyéb háztartási bútorok.
- **Könyvek:** különböző méretű és elrendezésű könyvek.

A képek elkészítése során figyeltünk arra, hogy azok különböző környezeti tényezőket tükrözzenek, mint például változó háttér, fényviszonyok és perspektívák. Ez a sokféleség biztosítja, hogy a modell képes legyen kezelni a valós adatok változatosságát. Az elkészült adathalmaz egy privát Hugging Face repository-ban került tárolásra, ahol az adatok biztonságosan elérhetők. A projekt során egy read-only hozzáférési tokent használtunk az adatok kezeléséhez, amely megakadályozza az adatok véletlen módosítását vagy törlését.

2.1.2. Adatok felosztása

Az adatokat a modell tanítása és értékelése érdekében két részre osztottuk:

1. **Tréning adathalmaz (80%):** Ez az adathalmaz a modell tanítására szolgál. A cél az, hogy a modell minél többet tanuljon az osztályok közötti különbségekről.
2. **Teszt adathalmaz (20%):** Ez az adathalmaz a modell teljesítményének értékelésére szolgál. Itt nem alkalmaztunk adatmódosító műveleteket, hogy a teszteredmények tükrözzék a modell valódi képességeit.

A két halmaz felosztását manuálisan ellenőriztük annak érdekében, hogy mindegyik osztály megfelelően legyen reprezentálva mind a tréning, mind a teszt adatokban.

2.1.3. Adatok előfeldolgozása

Az előfeldolgozás fő célja az volt, hogy a képeket olyan formátumra és minőségre hozzuk, amely ideális a modell számára, miközben minimalizáljuk az emberi beavatkozás szükségességét. Az előfeldolgozási lépések a következők:

1. **Képméretezés:** A képeket egységesen **128x128 pixeles** méretre állítottuk be. Ez a méret elegendő részletességet biztosít az osztályozáshoz, ugyanakkor alacsonyan tartja a modell számítási igényét. A képméretezés során figyeltünk arra, hogy megőrizzük az objektumok arányait, hogy ne torzuljanak.
2. **Adat augmentáció:** Az augmentációt kizárólag a tréning adatokon végeztük el, hogy növeljük a modell által látott adatok változatosságát, ezzel csökkentve a túlilleszkedés esélyét. Az alkalmazott augmentációs technikák a következők voltak:
    - **Véletlenszerű zoomolás:** Az objektumok közelebbi vagy távolabbi nézetének szimulálása.
    - **Véletlenszerű forgatás:** Az objektumok különböző szögekből történő megjelenítése.
    - **Véletlenszerű tükrözés:** Az objektumok horizontális megfordítása.
3. **Normalizálás:** A képadatokat normalizáltuk, hogy a pixelértékek a [0,1] tartományba essenek, ezzel optimalizálva a modell tanulási folyamatát. A normalizáció segíti a gradiens-alapú tanulás stabilitását.
4. **Tisztítás és validálás:** Bár az adatok nagy része megfelelő volt, manuálisan ellenőriztük a hibás címkéket és az irreleváns képeket. Ez különösen fontos, mivel a rossz minőségű adatok jelentősen ronthatják a modell teljesítményét.

2.1.4. Teszt és validációs adatok

A validációs és teszt adatokon nem végeztünk augmentációt, hogy megőrizzük "tiszta" természetüket. Ez lehetővé teszi a modellünk teljesítményének pontos értékelését, mivel a teszt adatok valós környezetben készült új adatok szimulálására szolgálnak.

2.2. Modell

A házi feladatban használt neurális hálózati modell a **MobileNetV2** architektúrára épül, amely kifejezetten alacsony számítási erőforrás-igényű feladatokra van optimalizálva. Ez a modell ideális választás volt számunkra, hiszen a célunk egy könnyen használható, de hatékony képosztályozó rendszer létrehozása volt. Az alábbiakban részletesen bemutatjuk a választott modellarchitektúrát, a tervezési döntéseket és a tanítási folyamat főbb lépéseit.

2.2.1. MobileNetV2 Architektúra

A MobileNetV2 egy mély konvolúciós neurális hálózat, amelyet mobil és beágyazott rendszerekre terveztek. Főbb jellemzői:

1. **Invertált maradékblokkok:** Az architektúra egyik kulcseleme az invertált maradékblokk, amely csökkenti a számítási költséget, miközben fenntartja a modell pontosságát.
2. **Mélységre szűkített konvolúció:** Ez a technika lehetővé teszi a számítási igény csökkentését azáltal, hogy minden konvolúciós műveletet csatornánként hajt végre, majd egy pont-konvolúciós művelettel kombinálja az eredményeket.
3. **Alacsony paraméterszám:** A modell kevesebb paramétert használ, mint sok más, hasonló teljesítményű architektúra, ami gyorsabb számítást és kisebb memóriaigényt eredményez.

2.2.2. Testreszabás

A projekt céljához igazítva a MobileNetV2-n a következő módosításokat végeztük el:

1. **Kimeneti réteg:** Az alapmodell utolsó rétegét egy teljesen összekapcsolt rétegre cseréltük, amely három kimenetet biztosít, az osztályok (cipők, bútorok, könyvek) számára. Az aktivációs függvény itt a **softmax**, amely biztosítja, hogy az osztályok valószínűségi eloszlást adjanak ki.
2. **Előtanítás:** A modell előre tanított változatát használtuk fel kiindulópontként, amely az ImageNet adathalmazon tanult vizuális mintázatokat. Ez a megközelítés lehetővé tette a gyorsabb konvergenciát és jobb teljesítményt a kisebb adathalmazon.
3. **Finomhangolás:** Az előre tanított rétegeket részben „befagyasztottuk”, és csak a magasabb szintű rétegeket tanítottuk újra, hogy a modell alkalmazkodjon az új adatokhoz.

2.2.3. Tanítási folyamat

2.2.3.1. Optimalizálás

A modell tanítása során az **Adam optimizátort** használtuk, amely egy adaptív tanulási rátát alkalmazó algoritmus. Ez a módszer gyors konvergenciát és stabil tanulási folyamatot biztosít. A tanulási rátát 0,001 értékre állítottuk be, ami megfelelő egyensúlyt teremt a gyors tanulás és a stabil konvergencia között.

2.2.3.2. Loss függvény

A választott **cross-entropy loss** függvény ideális többosztályos osztályozási feladatokra. Ez a függvény arra ösztönzi a modellt, hogy maximalizálja a valószínűséget a megfelelő osztály számára, miközben minimalizálja a hibás osztályok valószínűségét.

2.2.3.3. Korai megállás

A tanítás során korai megállási feltételt alkalmaztuk, amely megakadályozza a modell túlilleszkedését. A tanítás megszakad, ha a validációs adatokon mért pontosság egy meghatározott számú epoch után nem javul tovább. Ez biztosította számunkra, hogy a modell ne tanulja túl az „train" adathalmaz sajátosságait.

2.2.3.4. Teljesítménykövetés

A tréning folyamán a modell teljesítményét folyamatosan nyomon követtük a következő mutatók segítségével:

1. **Pontosság (Accuracy):** Az osztályozási pontosságot mind a „training", mind a validációs adatokon monitoroztuk.
2. **Veszteség (Loss):** A veszteség csökkenésének mintázata segített megérteni a modell tanulási folyamatát.
3. **Konfúziós mátrix (Confusion Matrix):** A teszt adatok osztályozási eredményeit vizualizáltuk, hogy azonosítani tudjuk az osztályok közötti hibákat.

2.2.3.5. Integráció PyTorch Lightning és Weights & Biases segítségével

Feladatunk során a PyTorch Lightning keretrendszert használtuk a „training" és értékelési folyamat egyszerűsítésére. Emellett a Weights & Biases (W&B) eszközt felhasználtuk a teljesítménymetrikák automatikus nyomon követésére és vizualizálására. A W&B segítségével valós időben tudtuk monitorozni a tanulási folyamatot, és könnyen elemezhető grafikonokat generálhattunk.

2.2.3.6. Modell értékelése

A tanított modell jól teljesített a teszt adatokon, 85% körüli pontosságot ért el, ami a választott architektúrához és az adatminőséghez képest kiváló eredmény. A teljesítmény részletei az alábbiak szerint alakultak:

- A cipők és könyvek osztályozása kimagaslóan pontos volt, míg a bútorok esetében előfordultak kisebb hibák.
- A „training" és validációs veszteségek közötti kis eltérés azt mutatta, hogy a modell nem hajlamos túlilleszkedésre.

2.3. Hiperparaméterek optimalizálása

A neurális hálózatok teljesítménye jelentős mértékben függ a megfelelő hiperparaméterek megválasztásától, ezért a projekt során nagy hangsúlyt fektettünk a hiperparaméterek optimalizálására. Ugyan a projekt keretein belül manuális keresési módszert alkalmaztuk, ez a folyamat tudatos és jól strukturált döntésekre épült, figyelembe véve a modell architektúráját és az adathalmaz sajátosságait.

2.3.1. Fontos hiperparaméterek

Az alábbiakban felsorolt hiperparaméterek kerültek optimalizálásra, mivel ezek nagy hatással vannak a tanulási folyamatra és a modell teljesítményére:

2.3.1.1. Tanulási ráta (learning rate): A tanulási ráta az egyik legfontosabb hiperparaméter, mivel meghatározza, hogy a modell milyen lépésközökkel frissíti a súlyait. A túl magas tanulási ráta instabilitáshoz vezethet, míg a túl alacsony érték lassú konvergenciát eredményez.

- Az Adam optimizer alapértelmezett tanulási rátáját, **0,001**-et választottuk kiindulási értéknek.
- Az értéket 0,0001 és 0,01 közötti tartományban teszteltük és a legjobb eredményt az alapértelmezett értéknél tapasztaltuk.

2.3.1.2. Batch méret (batch size): A batch méret az egyszerre feldolgozott adatmennyiséget határozza meg. Nagyobb batch méretek gyorsabb feldolgozást tesznek lehetővé, de több memóriát igényelnek, a kisebb batch méretek pontosabb gradiens-becslést biztosítanak.

- A rendelkezésre álló erőforrások alapján **32-es batch méretet** választottunk, így biztosítva egyensúlyt a memóriahasználat és a tanulási stabilitás között.
- Alternatív batch méretek (16 és 64) tesztelésével megállapítottuk, hogy a 32-es érték eredményezte a legjobb teljesítményt.

2.3.1.3. Epochok száma: Az epochok száma azt határozza meg, hogy az adathalmazt hányszor dolgozza fel a modell a tanítás során. Túl sok epoch túlilleszkedést eredményezhet, míg túl kevés esetén a modell alulilleszkedik.

- A training-et korai megállás (early stopping) feltétellel egészítettük ki, amely automatikusan leállítotja a tanítást, ha a validációs pontosság egy adott számú epochon keresztül nem javult. Ez tipikusan 15-20 epoch körül állt meg.

2.3.1.4. Adat augmentáció paraméterei: Az augmentáció célja, hogy növelje az adathalmaz változatosságát, de az augmentáció mértéke és típusa szintén finomhangolást igényelt.

- **Zoom mértéke:** 0-10%-os véletlenszerű zoom-ot alkalmaztunk.
- **Forgatás:** A képek véletlenszerű elforgatását ±15 fokos tartományban végeztük el.
- **Tükrözés:** Horizontális tükrözést alkalmaztunk, amely segítette az objektumok felismerését különböző nézőpontokból.

2.3.1.5. Dropout arány: A dropout az overfitting elleni küzdelem egyik kulcsfontosságú eszköze. A projekt során 0,3-as dropout arányt használtunk az utolsó rejtett rétegben, amely a legjobb egyensúlyt biztosította az általánosítás és a modell kapacitása között.

2.3.2. Optimalizálási folyamat

A hiperparaméterek optimalizálását manuális keresési megközelítéssel végeztük el, ami egy iteratív folyamatot jelentett. Az egyes iterációk során az alábbi lépéseket hajtottuk végre:

1. **Induló értékek meghatározása:** Az alapértelmezett értékeket (pl. tanulási ráta: 0,001; batch méret: 32) választottuk kezdeti konfigurációnak.
2. **Egyes hiperparaméterek módosítása:** Csak egy-egy paraméter értékét változtattuk, miközben a többi változatlan maradt.
3. **Modellek tesztelése:** Az egyes konfigurációkat rövidebb tanulási ciklusokkal (3-5 epoch) teszteltük, hogy gyors visszajelzést kapjunk az adott beállítások hatékonyságáról.
4. **Legjobb konfiguráció kiválasztása:** Az iterációk során összegyűjtött eredmények alapján kiválasztottuk a legjobb kombinációt.

3. Eredmények és értékelés

A projekt eredményei a modell teljesítményének számszerűsítésére és vizuális értékelésére épültek. Az értékelési folyamat során a modell különböző metrikákon keresztül került vizsgálatra, amelyek tükrözik az osztályozó hatékonyságát és megbízhatóságát mind a „training”, mind a teszt adatokon. Az alábbiakban részletesen ismertetjük az eredményeket, az alkalmazott értékelési módszereket és a vizualizációkat.

3.1. Teljesítménymutatók

3.1.1. Pontosság (Accuracy)

A pontosság az egyik legfontosabb mérőszám az osztályozási feladatoknál, amely megmutatja, hogy a modell milyen arányban jósolta meg helyesen az osztályokat a teljes adathalmazhoz viszonyítva.

- **Tréning pontosság:** A modell a tanítási adathalmazon kiemelkedő, nagyjából 90%-os pontosságot ért el, ami azt mutatja, hogy jól megtanulta az osztályok közötti különbségeket.
- **Teszt pontosság:** A teszt adatokon elért körülbelül 85%-os pontosság jelzi, hogy a modell képes általánosítani új, eddig nem látott adatokon. Ez az eredmény megfelelő egy valós környezetben készített adathalmaz esetében, amely jelentős változatosságot tartalmaz.

3.1.2. Veszteség (Loss)

A veszteségi függvény értéke azt mutatja, hogy a modell milyen mértékben tévedett az egyes előrejelzéseknél. A training és validációs veszteséget külön is elemeztük:

- A training veszteség folyamatos csökkenést mutatott az epoch-ok során, ami a tanulási folyamat stabilitását igazolja.
- A validációs veszteség alacsony értéke és az összhang a training veszteséggel azt jelzi, hogy a modell nem mutat túlilleszkedést (overfitting-et).

3.1.3. Konfúziós mátrix

A konfúziós mátrix vizuálisan szemlélteti, hogy a modell mennyire pontosan osztályozta az egyes kategóriákat. Az alábbi megfigyelések születtek:

- **Cipők:** A cipők osztályozása volt a legpontosabb, a legtöbb tesztkép helyes besorolást kapott.
- **Könyvek:** A könyvek osztályozása szintén pontosnak bizonyult, de néhány esetben bútorként kerültek besorolásra.
- **Bútorok:** A bútorok osztályozásában több hiba fordult elő, ezek közül néhány könyvekhez vagy cipőkhöz lett tévesen hozzárendelve.

A mátrix elemzése alapján az osztályok közötti főbb hibák a bútorok esetében fordultak elő, amely arra utal, hogy ez az osztály nehezebben megkülönböztethető a modell számára. Ez részben a vizuális hasonlóságoknak és a háttérvariációknak köszönhető.

3.2. Vizualizációk

3.2.1. Tréninggörbék

A veszteség és pontosság változásának vizualizációja az epoch-ok függvényében segített megérteni a tanulási folyamat dinamikáját:

- A training veszteség csökkenése azt mutatta, hogy a modell fokozatosan javította az előrejelzéseit.
- A validációs veszteség stabilizálódása és alacsony értéke azt jelezte, hogy a modell nem csak a tanulási adathalmazra specializálódott, képes általánosítani is.

3.2.2. Adatminták vizualizálása

A modell működésének megértését segítette az adatok vizualizálása:

- **Augmentált training adatok:** Véletlenszerű mintákat mutattak be az augmentált trainning adatokból, amelyek megmutatták, hogy az augmentáció hogyan járult hozzá a modell változatosabb mintákon történő tanulásához.
- **Teszt adatok előrejelzése:** A modell által a teszt adatokra adott előrejelzéseket vizuálisan is ellenőriztük. Az előrejelzések túlnyomó többsége helyes volt, de néhány téves besorolás is megjelent, különösen a bútorok esetében.

3.2.3. Összehasonlítás baseline modellekkel

Két baseline modellt használtunk összehasonlítási alapként:

1. **Véletlen osztályozó:** A véletlen osztályozás várható pontossága a három osztály esetében nagyjából 33% volt.
2. **Leggyakoribb osztály választása:** Ez a modell mindig a leggyakoribb osztályt (cipők) jósolta, amely körülbelül 45%-os pontosságot eredményezett.

A MobileNetV2 alapú modell jelentősen felülmúlta mindkét baseline modellt, ezzel bizonyítva a tanítási és finomhangolási folyamat hatékonyságát.

3.2.4. Eredmények összegzése

Az eredmények alapján a projekt sikeresen demonstrálta egy valós környezetben készült képosztályozó rendszer fejlesztését. A modell teljesítménye az alábbi kulcsténye zőknek köszönhető:

- Gondos adatfeldolgozás és augmentáció.
- Megfelelő modellarchitektúra kiválasztása és finomhangolása.
- Hatékony hiperparaméter-optimalizálás.

Az értékelés során nyert tapasztalatok arra is rámutattak, hogy a modell javítható lenne további adatok bevonásával, illetve a bútorkategória finomabb jellemzésével. Ez egyértelműen jelzi, hogy a projekt további fejlesztési lehetőségeket tartogat.