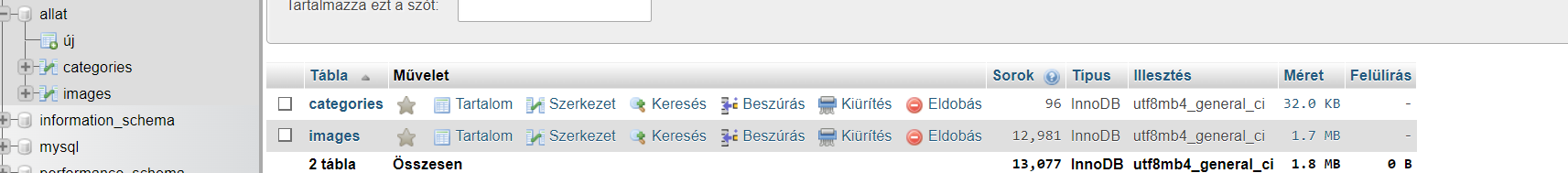
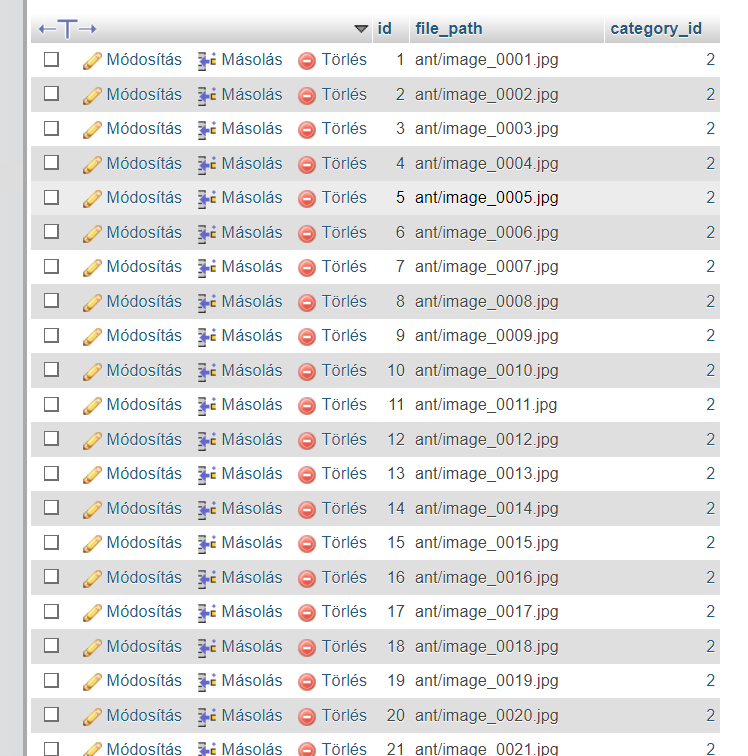
Állat Felismerő

(Csomor István Máté: I5V6ZM, Károly Zoltán Krisztián: JANT9G)

1. **Adatbázis**

Az adatbázist XAMPP MySQL-ben terveztük/hoztuk létre. Az adatbázis tartalmazza a kategóriákat, ami az állatok neveit tartalmazza az images pedig több mint 10000 képet tartalmaz ezt használtuk a betanítási folyamata alatt.

Mivel MySQL nem képes képek tárolására így a képek elérési útvonalait adtuk meg nekik. Mindegyik képet hozzá csatoltuk egy állat kategoríához, így a modell a betanítási folyamat alatt képes volt kategorizálni az állatokat.

1. **Modell Tanítás**

1. Főbb lépések

a. Adatbázis-kezelés

Adatbázis csatlakozás:

Kapcsolat létrehozása egy MySQL adatbázissal, amely tartalmazza a képek kategóriáit és az egyes képekhez tartozó metaadatokat.

Kategóriák száma:

Lekérdezés az adatbázisból, amely meghatározza a kategóriák számát.

Érvénytelen adatok eltávolítása:

Azok a képek, amelyek kategóriája nem található az adatbázisban, törlésre kerülnek.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírásAdatok lekérése:

A képek elérési útvonalai és azok kategóriái betöltődnek.

b. Adatok előkészítése

Képek feldolgozása:

A képek méretezése 128x128 pixelre történik.

A képek normalizálása: az értékek [0, 1] közé skálázódnak.

A kategóriákhoz tartozó címkék (labels) one-hot encoding formátumban kerülnek mentésre.

Adathalmaz szétválasztása:

A képek 80%-a a betanításhoz, 20%-a a validációhoz kerül.

Adatbővítés (data augmentation):

Forgatás, nagyítás, eltolás, tükrözés stb. révén mesterségesen növelik a betanításhoz használt adatok variabilitását.

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

c. Modell építése és betanítása

MobileNetV2 használata:

A modell a MobileNetV2 előre betanított hálózatot használja alapként.

Az utolsó rétegeket eltávolítják és a feladathoz testreszabják.

Rétegek módosítása:

Átlagos pooling réteg (GlobalAveragePooling2D) és teljesen összekötött rétegek (Dense) kerülnek a kimeneti réteg elé.

Betanítási beállítások:

Az alapmodell rétegei nem taníthatók (fagyasztva vannak).

Az optimalizáláshoz az Adam algoritmus kerül alkalmazásra.

Betanítás:

A modell 10 epochon keresztül tanul, a betanítás során a validációs halmazzal ellenőrzik a teljesítményt.

Modell mentése:

A betanított modell súlyai (weights) mentésre kerülnek egy fájlba.

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

d. Állatfelismerés funkció

Kép betöltése és előkészítése:

A megadott képet átméretezi 128x128 pixeles méretre és normalizálja.

Predikció:

A kép alapján a modell megállapítja a legvalószínűbb kategóriát.

Kategória nevének megjelenítése:

A kategória nevét az adatbázisból kérdezi le, és a végeredményt visszaadja.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

2. Felhasználási példa

Modell betanítása:

A kód automatikusan betölti az adatokat, előkészíti őket, majd a modellt betanítja a képek alapján.

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Állatfelismerés:

A detect\_and\_recognize\_animal függvény egy képről meghatározza, milyen állat látható rajta, és visszaadja annak nevét.

A képen szöveg, Betűtípus, képernyőkép látható

Automatikusan generált leírás

3. Hibakezelés

Nem létező vagy hibás képek automatikusan kihagyásra kerülnek.

Érvénytelen kategóriák kiszűrése adatbázis műveletekkel.

Kép betöltésekor fellépő hibák try-except blokkokkal kezeltek.

4. Jövőbeli fejlesztési lehetőségek

Valós idejű felismerés: Kameráról érkező képek közvetlen feldolgozása.

Kategóriák bővítése: Új állatok hozzáadása az adatbázishoz és a modell továbbtanítása.

Teljesítmény optimalizálás: Az alapmodell további rétegeinek taníthatóvá tétele.

1. **GUI**

1. Főbb lépések

a. Adatbázis-kezelés

Kapcsolódás az adatbázishoz:

A kód egy MySQL adatbázist használ, amely tartalmazza az állatkategóriákat és azok nevét.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Kategóriák lekérése:

Lekérdezi a kategóriák számát az adatbázisból, hogy a modell kimeneti rétegét ennek megfelelően konfigurálja.

b. Modell betöltése

MobileNetV2 alapmodell használata:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

A MobileNetV2 előre betanított neurális hálózatát használja alapként.

Az utolsó rétegek lecserélésre kerülnek a feladat specifikus kimeneti réteggel.

Súlyok betöltése:

A modell betanított súlyai a animal\_recognition\_model.weights.h5 fájlból kerülnek betöltésre.

Modell konfigurálása:

Az Adam optimalizálót és categorical\_crossentropy veszteségfüggvényt használja.

c. Állatfelismerési funkció

Kép feldolgozása:

A kiválasztott kép átméretezése 128x128 pixelre.

A kép normalizálása (értékek [0,1] közé skálázása).

Előrejelzés:

A modell predikciót ad, amely megadja a legvalószínűbb állatkategóriát.

Eredmény lekérése:

A felismerés alapján az adatbázisból lekéri az állat nevét.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

d. Felhasználói felület (Tkinter GUI)

Fő funkciók:

Kép tallózása: A felhasználó fájldialógus segítségével kiválaszthat egy képet.

Kép felismerése: A modell segítségével megjósolja, hogy milyen állat látható a képen.

Kép megjelenítése: A kiválasztott kép kicsinyített változatának megjelenítése egy vásznon.

Felhasználói elemek:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Szövegek és gombok: Az alkalmazás információkat és vezérlőelemeket tartalmaz a felhasználók számára.

Képmegjelenítő vászon: A kép megjelenítésére szolgáló terület.

GUI szerkezete:

Főablak: Cím, képválasztási lehetőség, felismerés indítása gomb, eredmény megjelenítése.

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

2. Hibakezelés

Hibás vagy hiányzó fájl: A program ellenőrzi, hogy a kiválasztott fájl létezik-e, és megfelelő-e a formátuma.

Adatbázis hibák: Az adatbázisból hiányzó kategóriák esetén hibaüzenet jelenik meg.

Általános kivételek: A try-except blokk biztosítja, hogy a kód futása ne álljon le váratlan hibák miatt.

3. Jövőbeli fejlesztési lehetőségek

Több nyelv támogatása: Az alkalmazás többnyelvűvé tétele.

Valós idejű felismerés: Kamera képének valós idejű feldolgozása.

Offline működés: Az adatbázis eltávolítása, és az állatkategóriák statikus fájlból való betöltése.

Modell továbbfejlesztése: A modell további állatkategóriákra való betanítása.