

# Gombaosztályozó feladat ResNet50 és EfficientNet alapján

**Bologa Eduárd**

**Kiss Hunor**

**Kozma Szabolcs**

**Pümkösti Györk**

## 1. Bevezetés

A projekt célja egy képosztályozó modell fejlesztése, amely képes azonosítani és kategorizálni különböző gombafajokat. A feladat során 100 különböző gombafaj képeit használtuk fel, és két különböző mély tanulási modellt alkalmaztunk: a ResNet50 és az EfficientNet architektúrákat. Az első próbálkozás során a ResNet50-t használtuk, majd a modell finomításával és optimalizálásával áttértünk az EfficientNet-re, hogy jobban alkalmazkodjunk a feladathoz. A fejlesztési folyamat során számos kihívással szembesültünk, amit különböző optimalizációs technikák alkalmazásával igyekeztünk megoldani, hogy javítsuk a modellek teljesítményét.

A dokumentáció célja, hogy részletesen bemutassa az alkalmazott modelleket, a felmerült problémákat és azok megoldásait, valamint a modellek teljesítményét és az elért eredményeket. Az alábbiakban ismertetjük a régi ResNet50 modellt, az új ResNet50 implementációját és az EfficientNet modellt, végül pedig az egyes modellekhez tartozó eredményeket és azok elemzését.

## 2. Régi ResNet50 Modell

### 2.1. Problémák és nehézségek

Első lépésként a hagyományos ResNet50 architektúrát alkalmaztuk a gombaosztályozásra. Az adatokat egy zip fájlból, közvetlenül olvastuk be, ami nagyon lassú képbetöltési időt biztosított a modell számára. Azonban, annak ellenére, hogy a képek gyorsabban betöltődtek, a modell eredményei elmaradtak a várakozásoktól. A validációs pontosság folyamatosan ingadozott 0 és 8 százalék között, ami arra utalt, hogy a modell nem volt képes megfelelően azonosítani a gombafajokat.

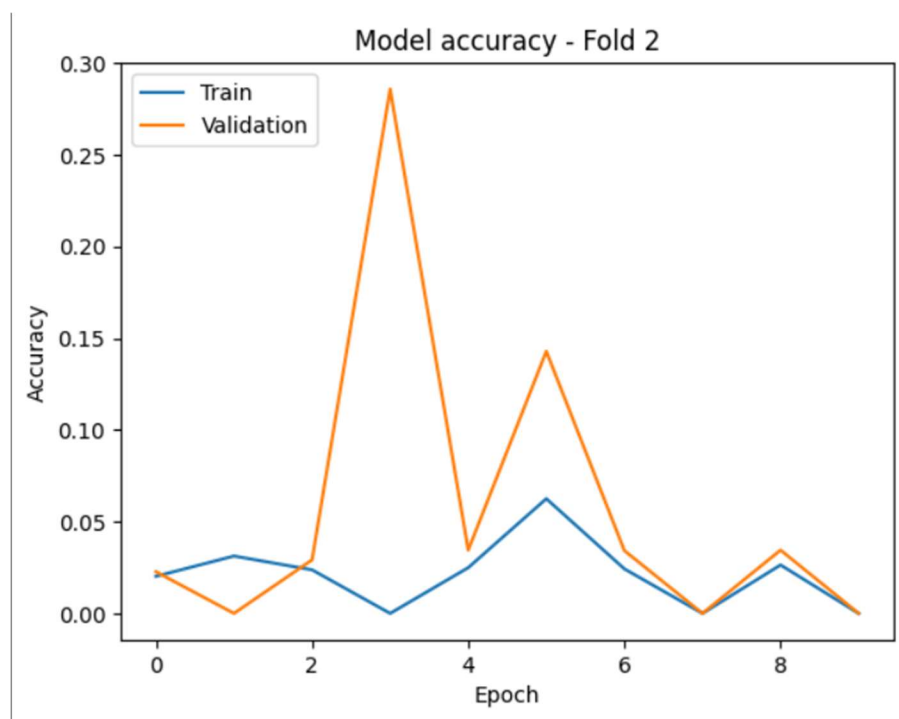
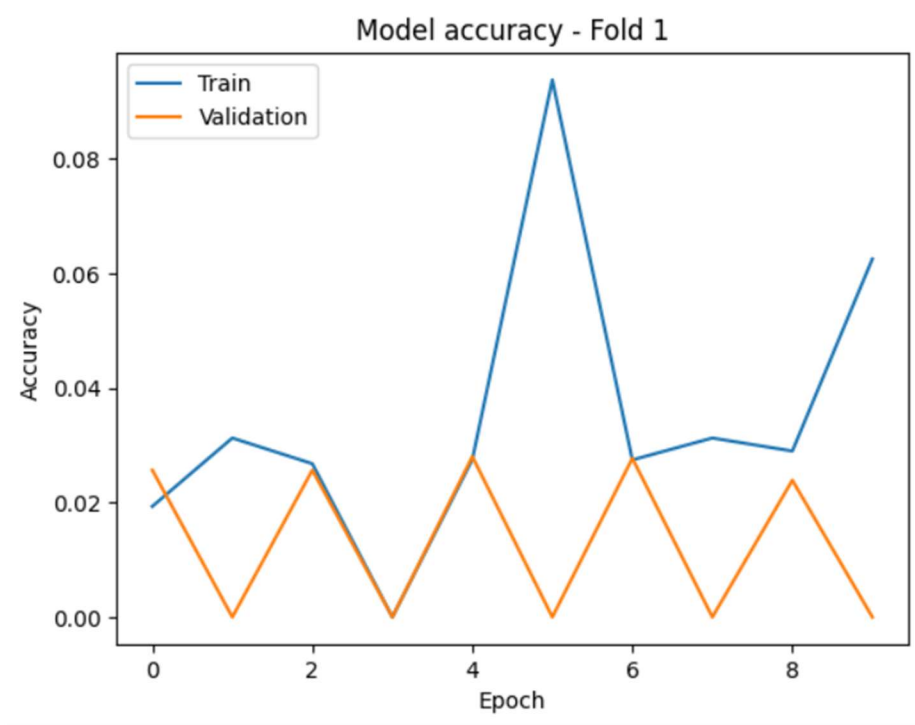
A tréning során nem láttunk jelentős fejlődést a modell teljesítményében, ami arra engedett következtetni, hogy a hálózat nem tanulta meg megfelelően a gombák jellemzőit. További problémák is felmerültek: a Google Colab környezetben a tanítás lassú volt, és gyakran megszakadt. Ha a modell nem tudta betölteni az összes képet a memóriába, akkor a RAM-korlátozások miatt a tanulás megszakadt, és az egész tanítási folyamat újraindításra kényszerült.

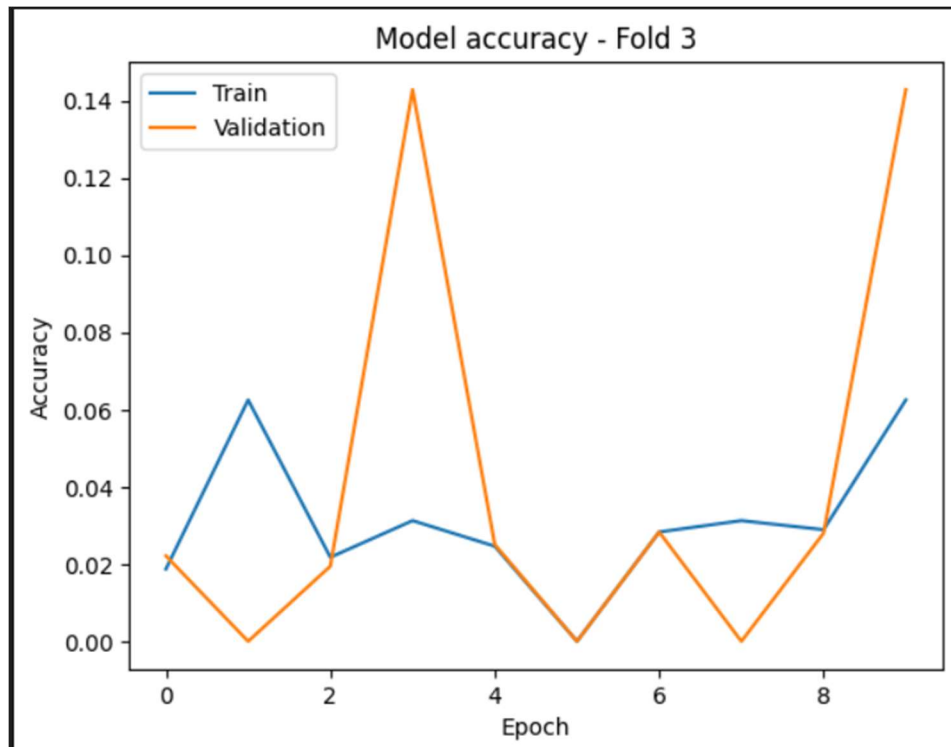
### 2.2. Képbetöltési Folyamat

A régi modellnél az **elérési utak** használata mellett a képek betöltése és feldolgozása időigényes volt, mivel minden egyes képet újra megnyitottunk és bezártunk, ami jelentősen lelassította a folyamatot. Mivel a gombafajok közötti különbségek finomak lehetnek, az adatok precíz feldolgozása rendkívül fontos, de a modell nem tudta ezt a lépést megfelelően kezelni.

### 2.3. Konklúzió

Összességében nézve ez a megközelítés nem vont maga után nagyobb sikert, hiszen a kód nehézkesen fodult le, és szinte semmit nem tanult a rendszer.





## 3. Új ResNet50 modell

### 3.1. Új Implementációs Stratégia

Miután a régi ResNet50 modell nem hozott megfelelő eredményeket, úgy döntöttünk, hogy újrainplementáljuk a modellt TensorFlow alapú megoldásban, és eltávolítjuk a PyTorch-ot. A célunk az volt, hogy a tanítási folyamat gyorsabbá és stabilabbá váljon, miközben a gombaosztályozás specifikus igényeire is jobban reagálunk.

### 3.2. Képbetöltési Optimalizálás

Az új implementáció során a képek betöltésére az **OpenCV** könyvtárat használtuk. Ez lehetővé tette számunkra, hogy gyorsan és hatékonyan töltsük be a gombafajok képeit. Az adatfeldolgozási lépéseknél az **arány megtartására** helyeztük a hangsúlyt, így a képek nem torzulnak el, és a fontos jellemzők nem vesznek el az átméretezéskor. A **padding** használata biztosította, hogy a képek megtartsák a megfelelő arányokat, ami javította a modell tanulási teljesítményét.

### 3.3. Adatfeldolgozás és Loaddata Függvény

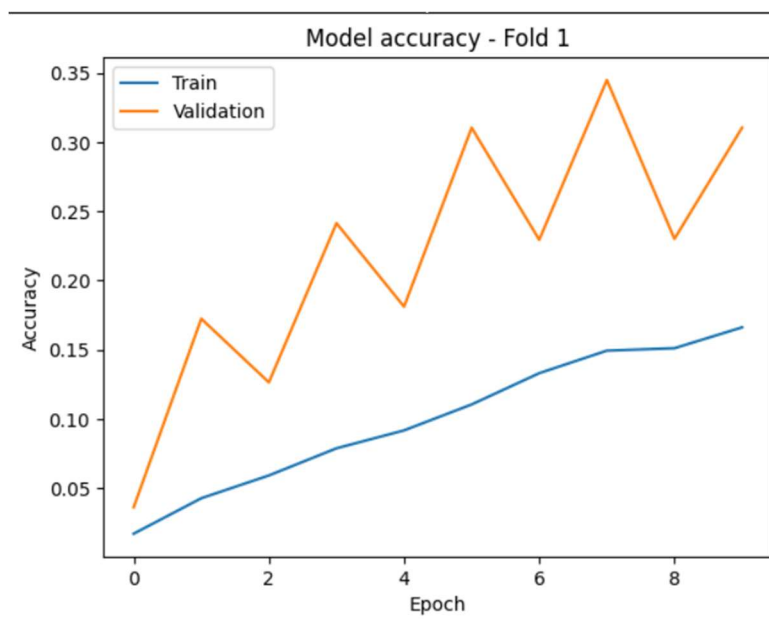
A Loaddata függvény kulcsszerepet játszott az adatok előkészítésében, mivel lehetővé tette a képek és címkék gyors és hatékony kezelését. A címkéket one-hot enkódoltuk, hogy a modell könnyen kezelhesse az osztályokat. Az adatok felosztása érdekében KFold kereszt-validációt alkalmaztunk, amely biztosította, hogy minden egyes adatot különböző mintákra bontottunk, így a tanulás nem a tanítóadatokhoz illeszkedett túlzottan.

### 3.4. A Modell Architektúrája

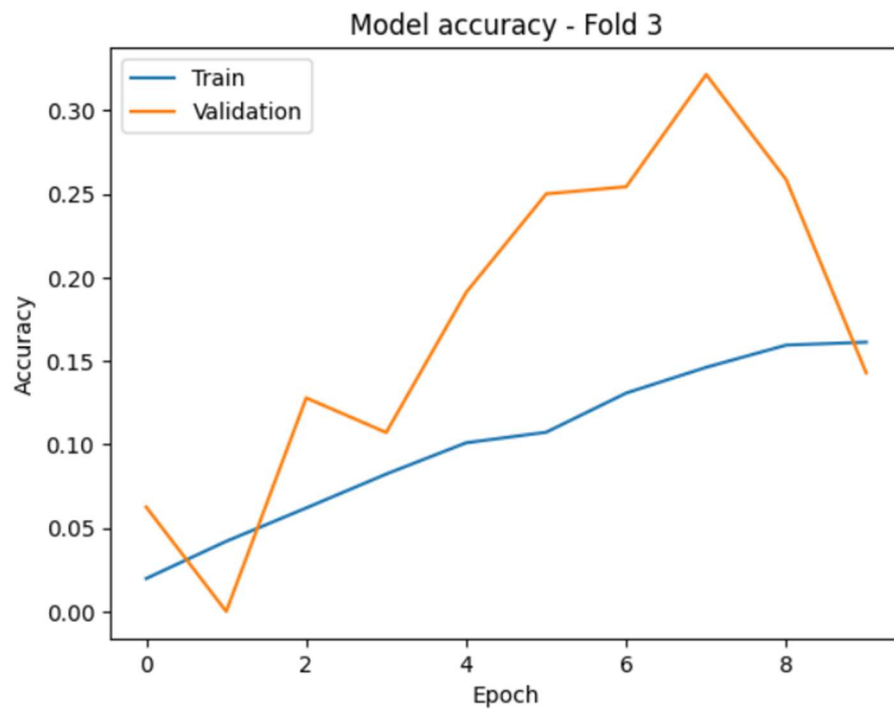
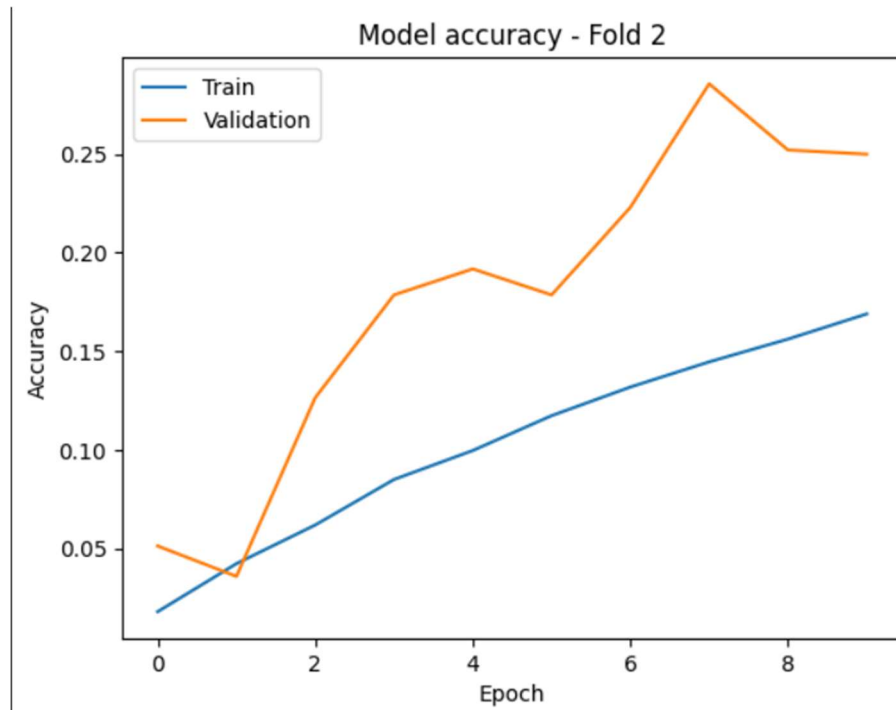
A ResNet50 modell fejének helyére AveragePooling réteget alkalmaztunk, amely csökkentette a modell komplexitását, miközben megtartotta a fontos információkat. A képek vektorizálása után két lineáris réteget alkalmaztunk, mindegyikben 256 neuronnal. A túltanulás (overfitting) elkerülése érdekében Dropout rétegeket is hozzáadtunk a modellhez. A tesztelés során azt tapasztaltuk, hogy a két 256-os réteg megfelelő eredményeket hozott, de úgy gondoltuk, hogy több réteg hozzáadásával hosszú távon javulhatnak az eredmények.

### 3.5. Metrikák és Eredmények

A tanulás során folyamatosan nyomon követtük a **pontosságot** és a **veszteséget**, és az eredmények alapján a validációs pontosság stabilabbá vált. Az új ResNet50 modell képes volt



megfelelően tanulni a gombák jellemzőiből, és a gombaosztályozás pontossága is javult. A modellezési folyamat gyorsabb lett, és jobban alkalmazkodott a Google Colab környezethez.



## 4. Új EfficientNet Modell

### 4.1. Különbségek és Fejlesztés

A EfficientNet modell egy továbbfejlesztett változata a hagyományos ResNet architektúráknak. Az EfficientNet automatikusan optimalizálja a hálózati rétegeket, és kevesebb számítási erőforrást igényel, miközben jobb teljesítményt biztosít. A gombaosztályozás során az EfficientNet még hatékonyabban képes volt felismerni a gombafajokat, miközben csökkentette a paraméterek számát.

### 4.2. Eredmények

Az EfficientNet modell tesztelése során az alábbi eredményeket kaptuk:

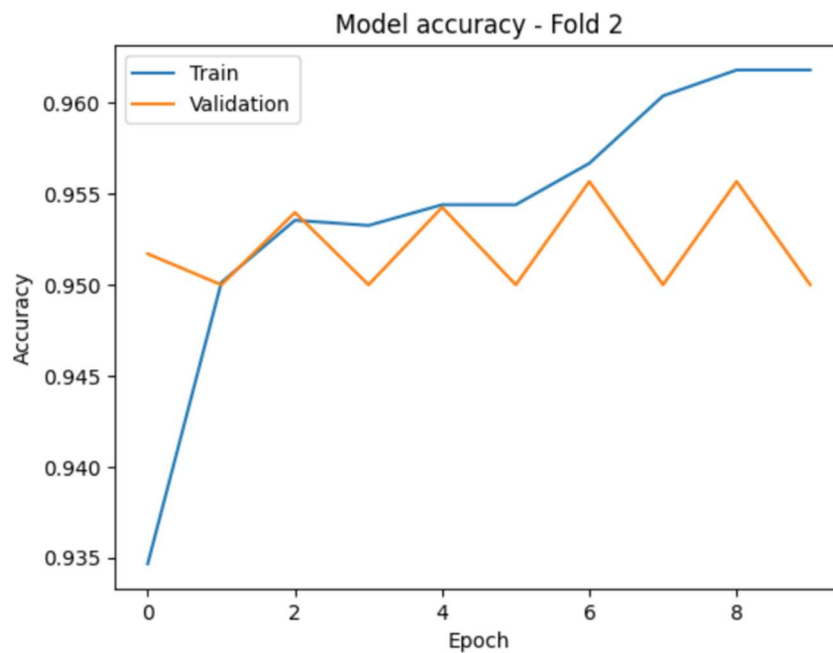
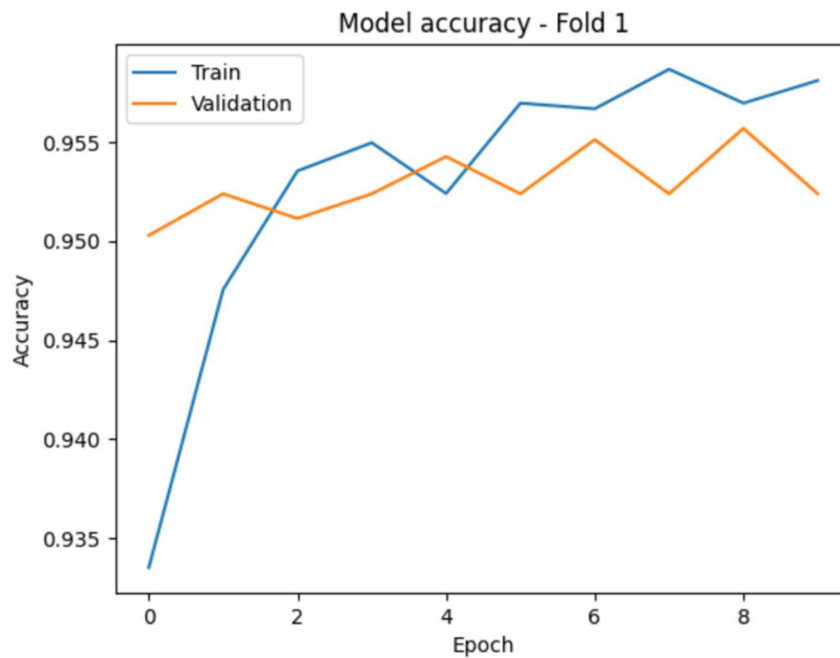
- Az EfficientNet magasabb validációs pontosságot ért el, mint a ResNet50.
- Kevesebb memóriahasználatot igényelt, amely különösen hasznos volt a Google Colab környezetben, ahol a memória erőforrások korlátozottak.
- Az EfficientNet gyorsabban konvergált a tanulás során, és jobb eredményeket produkált, mivel hatékonyabban használta fel az erőforrásokat.

A modell teljesítménye az alábbi két szélsőséges esetben mutatta meg a legjobban és legrosszabban a képességeit:

- Legjobb Eredmény: A legjobb teljesítményt az hozta, amikor a modell 96.20%-os pontosságot ért el a tanításhoz használt adatokon, és a validációs pontosság 95.00%-ra emelkedett. Az elvesztett érték (loss) 0.1110 volt, míg a validációs veszteség 0.0888. Ez a csúcserték azt jelzi, hogy a modell sikeresen alkalmazkodott a gombafajok osztályozásához, és jól generalizált az ismeretlen, validációs adatokra is.
- Legrosszabb Eredmény: A leggyengébb teljesítmény akkor jelentkezett, amikor a modell 92.23%-os pontosságot ért el a tanításhoz használt adatokon, míg a validációs pontosság 95.03% volt. A veszteség értéke 0.3860, a validációs veszteség pedig 0.1870. Bár a kezdeti

eredmények nem voltak kiemelkedően alacsonyak, a modell még nem volt kellően finomhangolva, hogy elérje a legjobb teljesítményt.

Ezek az eredmények azt mutatják, hogy az EfficientNet sikeresen alkalmazható a gombaosztályozásban, és képes nagy pontossággal kezelni a komplex, 100 gombafajból álló adatokat, miközben fenntartja a gyors tanulást és az optimális számítási erőforrás-használatot.





## 5. Eredmények és Összegzés

### 5.1 Modellek Eredményei

A tesztelés és az elemzés alapján a három modell alábbi eredményeket hozta:

- **Régi ResNet50:** Az eredmények ingadoztak, és a tanulás nem volt stabil.
- **Új ResNet50:** Az új modell stabilabbá vált, és a gombok felismerésében jelentős javulást eredményezett.
- **EfficientNet:** Az EfficientNet a legjobb eredményeket hozta, mind a pontosság, mind a memóriahasználat szempontjából.

### 5.2 Colab Hibák

A Google Colab környezetben a memóriahibák és a tanulási folyamat megszakadása továbbra is problémát jelentettek. Azonban a **modell optimalizálása és a memóriahasználat javítása** révén az új modellek már képesek voltak jobban kezelni a rendelkezésre álló erőforrásokat.

### 5.3. Összegzés

Összességében a feladat nagyon érdekes és hasznos gyakorlást képes nyújtani az adatelemzésbe, amin keresztül rengeteget lehet tanulni a mélytanulás területén.