A képen szöveg, Betűtípus, képernyőkép, embléma látható

Automatikusan generált leírás

Mesterséges intelligencia

**Pytorch Image Classification**

**Deep Learning-el**

**Foodifier**

Ételtípusok felismerése képről a Food 101 dataset alapján

**Kálmán Dávid**

**LNF8KS**

2023/2024/2

Tartalom

[Használati útmutató 3](#_Toc161995034)

[Telepítési útmutató 3](#_Toc161995035)

[Használata fájlból 3](#_Toc161995036)

[Használata weboldalon keresztül 3](#_Toc161995037)

[Probléma bemutatás 4](#_Toc161995038)

[Használt könyvtárak, technológiák, adathalmazok 5](#_Toc161995039)

[PyTorch 5](#_Toc161995040)

[fastai 5](#_Toc161995041)

[Food 101 Dataset 5](#_Toc161995042)

[Források 6](#_Toc161995043)

## Használati útmutató

A projekt során használt GitHub repository: <https://github.com/davidkalmn/image_classification_food_MI.git>

### Telepítési útmutató

1. Telepítsük a számítógépre a Python 3.11-es verzióját (a PyTorch Windowson csak a 3.8 és 3.11 közötti verziókat támogatja, így célszerű ezek közül a legfrissebbet használni a hibák elkerülése végett).
2. Klónozzuk a GitHub repository-t a kívánt célmappába terminálon keresztül:
   * *git clone https://github.com/davidkalmn/image\_classification\_food\_MI.git*
3. Telepítsük a szükséges csomagokat terminálon keresztül:
   * *pip install torch torchvision torchaudio*
   * *pip install -Uqq fastai*

### Használata fájlból

Ha lokálisan, fájlból szeretnénk használni a Foodifier-t:

1. A klónozott git mappán belül keressük meg a ***use\_food\_model.ipynb*** fájlt (a ***/use\_model*** mappán belül található), majd nyissuk meg Visual Studio Code-al, vagy valamilyen ehhez hasonló fejlesztői környezettel.
2. A megnyitott Jupiter Notebook kiterjesztésű fájlban futtassuk le az első 2 kódblokkot, amelyek az Előkészületekhez tartoznak.
3. Az utolsó kódblokkban, a 3. sorban adjuk meg a *local\_image\_path* változó értékeként a tesztelésre szánt kép elérési útvonalát, majd futtassuk le a kódblokkot.

### Használata weboldalon keresztül

Ha a projekthez készített weboldalon keresztül szeretnénk használni a Foodifier-t:

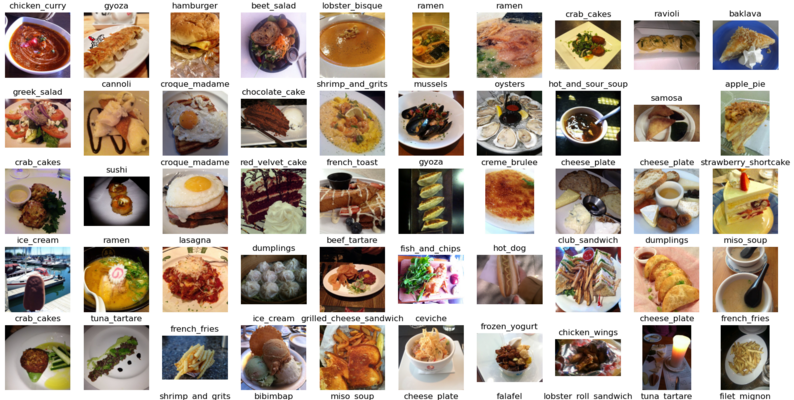
1. Megy

## Probléma bemutatás

A hétköznapi életben gyakran találkozunk olyan ételekkel, amelyeket nem ismerünk fel, vagy csak nehezen tudunk azonosítani. Az éttermekben, kávézókban vagy akár saját konyhánkban is előfordulhat, hogy egy étel neve vagy összetevői rejtélyesek számunkra. Ez a program megoldást nyújthat erre a problémára, mivel lehetővé teszi a képeken szereplő ételek automatikus azonosítását és osztályozását.

A Food 101 dataset alapján kialakított deep learning modellünk segítségével könnyen meg tudjuk állapítani, hogy egy fotón milyen étel szerepel, tegyük ezt csak szórakozásból vagy tanulásból és önellenőrzésről.

A Food 101 dataset gyakran használatos az ételfotók automatikus osztályozásának és az ételfajták felismerésének a tesztelésére és tanítására deep learning modellek esetében. Emellett fontos eszköze lehet az ételfotók alapján történő automatikus elemzésnek és kategorizálásnak az élelmiszeriparban és a kutatásban.



*Részlet a Food 101 Dataset-ből*

## Használt könyvtárak, technológiák és adathalmazok

### PyTorch

A PyTorch egy gépi tanulási könyvtár, amely a Torch könyvtáron alapul, és olyan alkalmazásokhoz használható, mint a számítógépes látás, adatfeldolgozás és a nyelvfeldolgozás.

Eredetileg a Meta AI fejlesztette, jelenleg pedig a Linux Foundation ernyője alatt működik. A PyTorchot jelenleg az egyik legnépszerűbb gépi tanulási könyvtár a TensorFlow mellett. [1]

### fastai

A fastai egy Python alapú, deep learning könyvtár és tanulási környezet, amely a PyTorch keretrendszerre épül. Azért fejlesztették ki, hogy egyszerűsítse és felgyorsítsa a deep learning modellek fejlesztését és tanítását. A fastai széles körű eszközöket kínál az adatbetöltéstől és -feldolgozástól kezdve egészen a modellfinomításig és értékelésig.

Egyik fő jellemzője a fastai-nak az ún. "opininated defaults" megközelítés, ami azt jelenti, hogy egyfajta alapértelmezett beállításokat használ, amelyeket a fejlesztők úgy választottak ki, hogy jól működjenek a legtöbb esetben. Ezáltal a fastai egyszerűbbé és gyorsabbá teszi a deep learning modellek fejlesztését anélkül, hogy a felhasználónak mélyreható ismereteket kellene szereznie a hiperparaméterek és más részletek beállításában. [2]

### ResNet

A ResNet (Residual Neural Network) egy mély neuronhálózat (deep neural network) architektúra, amelyet a Microsoft Research fejlesztett ki. Az egyik legkiemelkedőbb jellemzője az úgynevezett "residual blocks" vagy "maradék blokkok" használata, amelyek segítenek abban, hogy a hálózatok mélyebb rétegekben is hatékonyan tanuljanak, anélkül, hogy a gradiensek eltűnnének vagy rosszul illeszkednének a hálózat más rétegeihez.

A ResNet architektúra alkalmazása a képfelismerési feladatokban, mint például az ételfotók osztályozása, nagyon előnyös. Ennek oka az, hogy a ResNet képes kivonni és kifejezni az összetett jellemzőket a képekben, így lehetővé téve a modellnek, hogy pontosan és hatékonyan különböztessen meg különböző ételtípusokat.

Emellett számos előre kiképzett ResNet modell elérhető, amelyeket könnyen alkalmazhatunk és finomíthatunk a specifikus feladatokhoz, mint például az ételfotók osztályozása (pl. ResNet-34, ResNet-50, ResNet-101).

### Food 101 Dataset

A Food 101 dataset egy adatkészlet Kaggle-ről, amelyet az élelmiszerek képeinek és azokhoz tartozó címkéknek a gyűjtésére és összeállítására használnak gépi tanulási és képfeldolgozási projektekben. Ez a dataset 101 különböző ételtípust tartalmaz, több mint 100 000 képpel, amelyek mindegyike egy adott ételtípust reprezentál.

Minden képen egy specifikus ételtípus található, például pizza, hamburger, sushi, gyümölcssaláta stb. Minden ételtípushoz legalább 100 kép tartozik. Az adatkészlet minősége kiváló, a képek magas felbontásúak és változatosak, különböző szögekből és körülmények között készültek. [3]

## A program bemutatása

## Források

[1]: <https://en.wikipedia.org/wiki/PyTorch>

[2]: <https://en.wikipedia.org/wiki/Fast.ai> & <https://www.fast.ai/>

[3]: <https://www.kaggle.com/datasets/dansbecker/food-101>

[4]: <https://en.wikipedia.org/wiki/Residual_neural_network>