

Sémantická segmentace

(sémantické) přiřazení třídy každému pixelu v obraze

Sémantická segmentace

- Když jsme segmentovali Luční, bylo v tom spousta proměnných, které byly nastaveny na konkrétní úlohu.
- Zde použijeme konvoluční neuronové sítě, která dokáže segmentaci určit pro každý pixel. Rozhoduje se na základě sémantiky – jaké pixely jsou okolo, jakou mají vlastnost, kolik jich je, atd. (pixel se stejnou hodnotou může být bez problému součástí více tříd)
- Je to tedy univerzálnější řešení. Potřebujeme k tomu ale více dat a výpočetního výkonu.
- Využíváme deeplabv3+ architekturu, a hodnoty ze sítě resnet-18 („ví jak vypadá svět“, viz transfer learning)
 - Je pak relativně snadné přeučit síť na vlastních datech pro segmentaci vlastních obrázků.

Příprava k úloze

- Součástí balíku je natrénovaná síť na datech z map.
 - Rozeznávají se budova, cesta, pole, les, tráva
- Dále skripty k pohodlnější práci se sémantickou segmentací (z [tohoto](#) příkladu)
- Když pustíte skript, měl by vypadnout segmentovaný kus mapy.

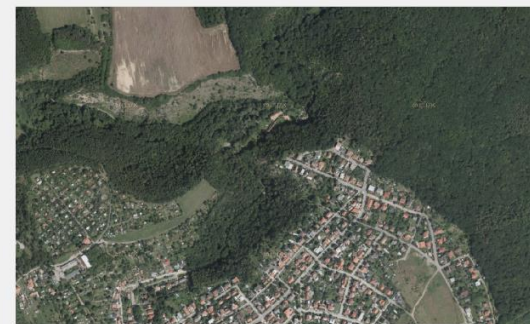


Image Labeler



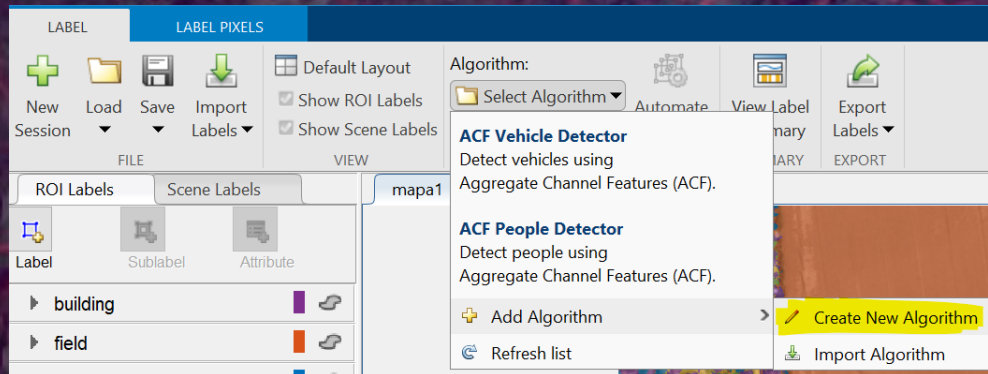
- Aplikace Matlabu sloužící k označení správných labelů například pro segmentaci. ([dokumentace](#)).
- V tomto případě budete používat Pixel Label (každý pixel je označen labelem (~barvou))
- Umožňuje ručně „pokreslit“ trénovací snímky patřičnou třídou.
 - Také uložit ve formátu, který snadno komunikuje s tréninkem neuronové sítě.
- Umožňuje též využít libovolný algoritmus k automatizaci procesu (člověk to poté už jen upraví)
 - Tvorba takového algoritmu je dnešní úloha

Úloha

- Co nejvíce si zjednoduší práci při labelování nových snímků.
- Vytvoříte algoritmus (použijte šablonu viz další stránka), který využije již natrénovanou síť k segmentaci obrazu (podrobně rozebráno na další stránce).
- Následně byste takto olabelované snímky uložily a síť přetrénováli.
 - Přetrénování = vezmete již naučenou síť a spustíte na ní trénink. Něco jako Transfer learning (používáte již naučenou síť), ale neměníte třídy, architekturu, atd, pouze zlepšujete schopnosti sítě.
- Tento cyklus se dá opakovat a síť neustále zlepšovat.

Úloha

- Spustíte příkaz imageLabeler, načtete soubor LabelsDefinition (Load -> Definitions).
- Vytvořte šablonu pro algoritmus (viz obr)
- Do šablony doplňte algoritmus, který načte neuronku a klasifikuje vstupní obrázek.
- Šablona je třída v matlabu. Je to dost podobné třídám z jiných jazyků s trochu odlišnou syntaxí.



Úloha

- Properties(constant) -> definice readonly properties -> nastavte jméno a popisek.
- Properties -> klasické měnitelné vlastnosti, zde vytvořte například proměnnou „net“.
- Methods -> Metody, do kterých budete doplňovat kód.
checkLabelDefinition -> Jestli jsou správně definované labely. Nemusíte tu nic, jen vrátit true . checkSetup -> to samé. Initialize -> zde přiřadíte proměnné „net“ příslušnou hodnotu (načtete síť)(provádíte to na objektu algObj, což je v C# známo jako „this“). Run -> tu provádíte vlastní segmentaci. Vracíte autolabels, musí být typu „categorical“.

Úloha

- Ověřte že váš algoritmus funguje tím, že načtete obrázek (v imageLabeleru) a pustíte na něj automatizaci.
- Odevzdáváte automatizační třídu (jeden .m soubor).