Fakulta agrobiologie,
potravinových a přírodních zdrojů

Digitalizace závlahových soustav z vodohospodářských map pomocí metod strojového učení

ING. ADAM TEJKL, PH.D.

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA

FAKULTA AGROBIOLOGIE, POTRAVINOVÝCH
A PŘÍRODNÍCH ZDROJŮ

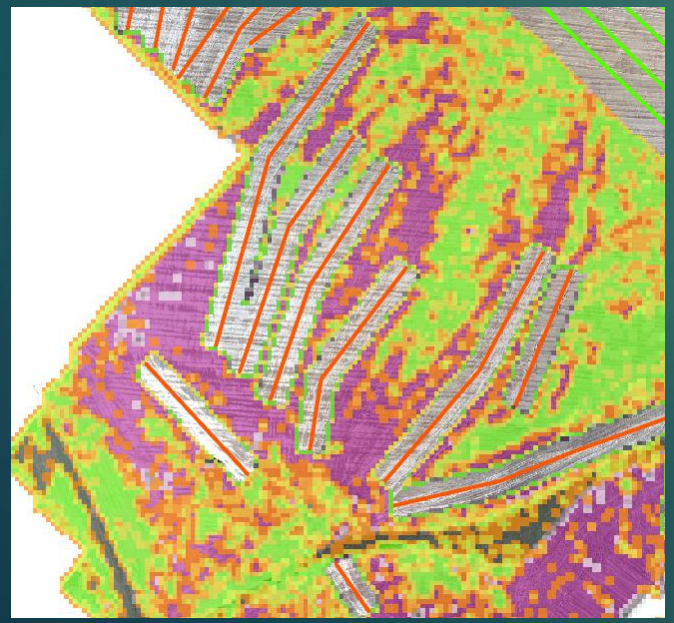
MZe 2025

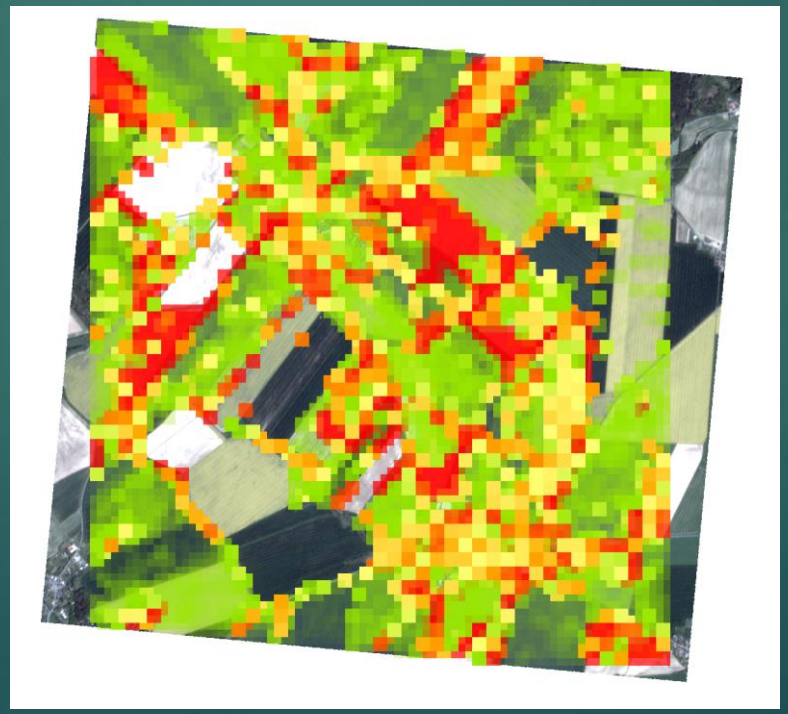
Motivace

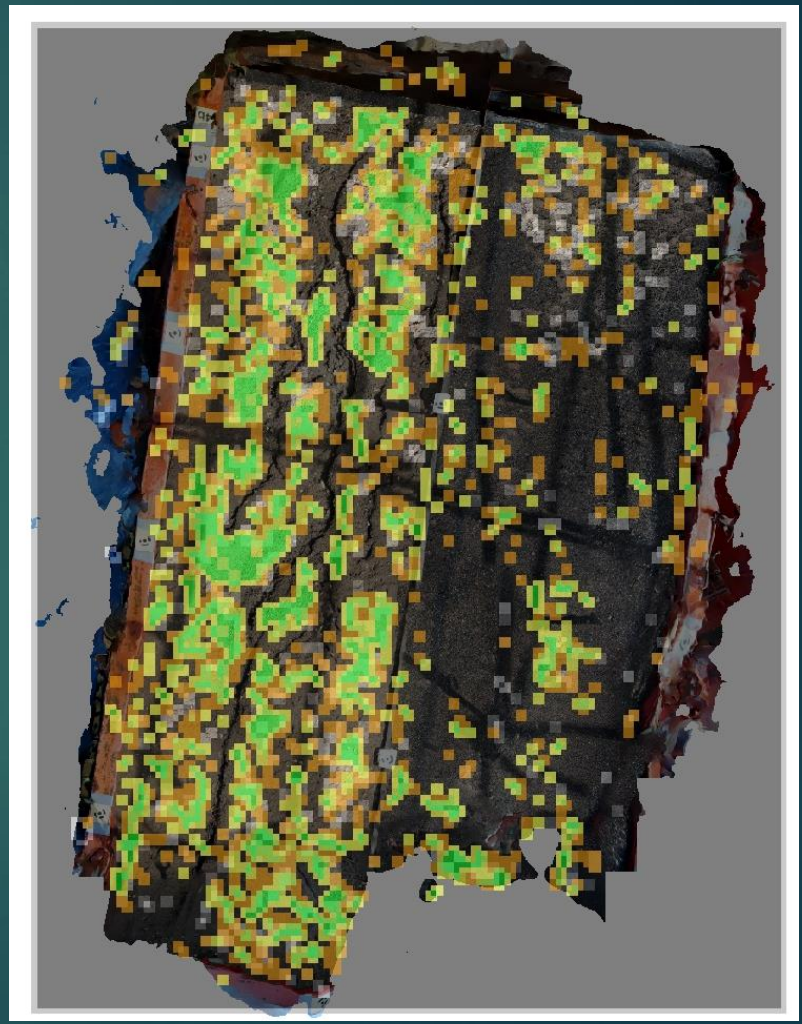
- Měnící se klima zvyšuje potřebu vody na krytí evapotranspirace
- Potřeba zvláště vysoká v zelinářských oblastech
- Rozsáhlé závlahové soustavy již vybudovány
- Data o rozsahu soustav nejsou vždy dostupná
- Základní Vodohospodářské Mapy zachycují podzemní síť potrubí závlahových soustav

Předchozí práce

- Klasifikace erozních rýh z ortofotosnímků
- Satelitní data
- Fotogrametrie
- Data z UAV







Cíle

- Identifikovat sítě závlahových soustav
- Identifikovat čerpací stanice závlahových soustav
- Vytvořit mapu polních bloků dle LPIS, s přivedenou závlahou
- Příprava dat a analýza výsledků v ArcGIS
- Jednoduše přetrénovatelný model

Metodika

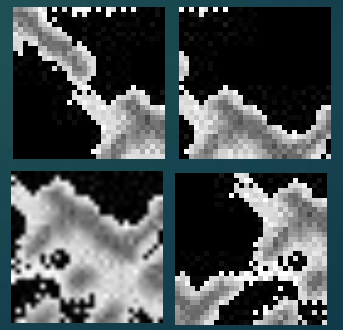
- Ruční tvorba trénovacích polygonů
- Segmentace mapových listů

- Trénink modelu v PyScripteru
- Použití modelu na zbytku dat v PyScripteru
- Import klasifikovaných listů do ArcGIS

- Další statistické analýzy

Posun segmentu

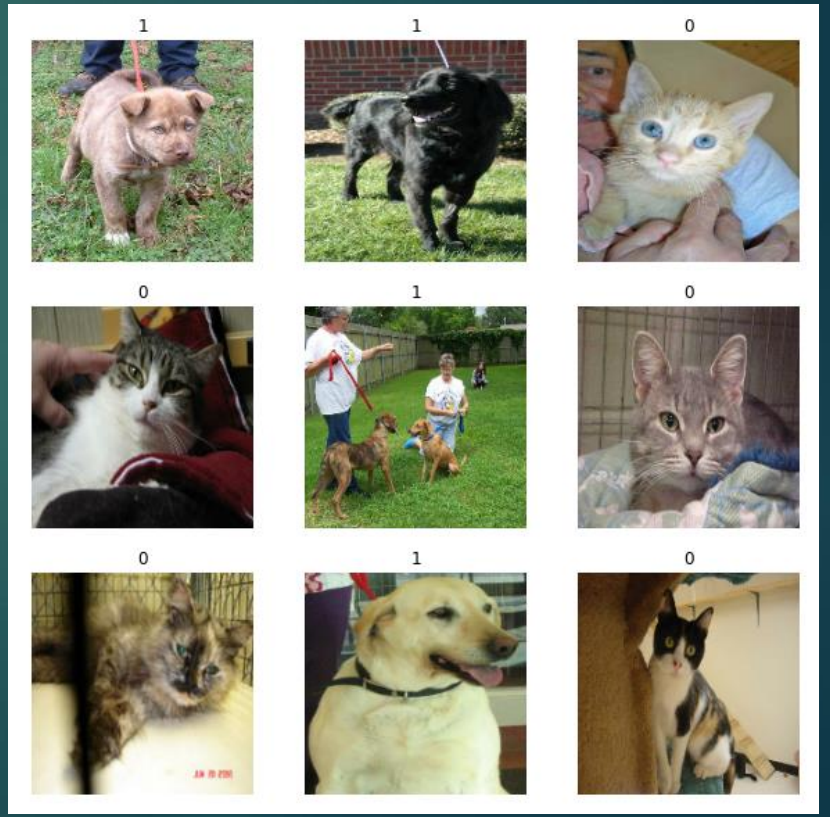
- Úspěšnost klasifikace závisí na dostatku trénovacích dat
- 4 sady dat jsou tvořeny z jednoho mapového listu
- Zvýšení množství trénovacích dat téměř na čtyřnásobek
- Zvýšení prostorového rozlišení klasifikace na dvojnásobek
- Prostorové rozlišení klasifikace = Délka hrany segmentu v pixelech



Model

- Python, Tensorflow, Keras
- Deep learning API
- Flexibilní, Jednoduchý, Výkonný
- Používán NASA, YouTube, Waymo (Chollet 2021)

- Kaggle Cats vs Dogs použit jako základ
- 23 410 snímků koček a psů
- Binární klasifikace



Fakulta agrobiologie, potravinových a přírodních zdrojů

Model

```python
#----------------------------------------------------------------------------
# Name:        module1
# Purpose:
#
# Author:      Adam Tejkl
#
# Created:     05.08.2021
# Copyright:   (c) Adam Tejkl 2021
# Licence:     <your licence>
#----------------------------------------------------------------------------
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

import os

print("Import of libraries done")

#%%
adress = "S:/Private/_PROJEKTY/2020_TACR_DPZ/mapa_udalosti/Tejkl_reseni/mosaics"

num_skipped = 0
for folder_name in ("NoRill", "Rill"):
    folder_path = os.path.join(adress, folder_name)
    for fname in os.listdir(folder_path):
        fpath = os.path.join(folder_path, fname)
        try:
            fobj = open(fpath, "rb")
            is_jfif = tf.compat.as_bytes("JFIF") in fobj.peek(10)
        finally:
            fobj.close()

        if not is_jfif:
            num_skipped += 1
            # Delete corrupted image
            os.remove(fpath)

print("Deleted %d images" % num_skipped)

#%%

image_size = (33, 132)
batch_size = 32

train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    adress,
    validation_split=0.2,
    subset="training",
    seed=1337,
    image_size=image_size,
    batch_size=batch_size,
)

val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    adress,
    validation_split=0.2,
    subset="validation",
    seed=1337,
    image_size=image_size,
    batch_size=batch_size,
)

print("Training and validation done")

#%%
```

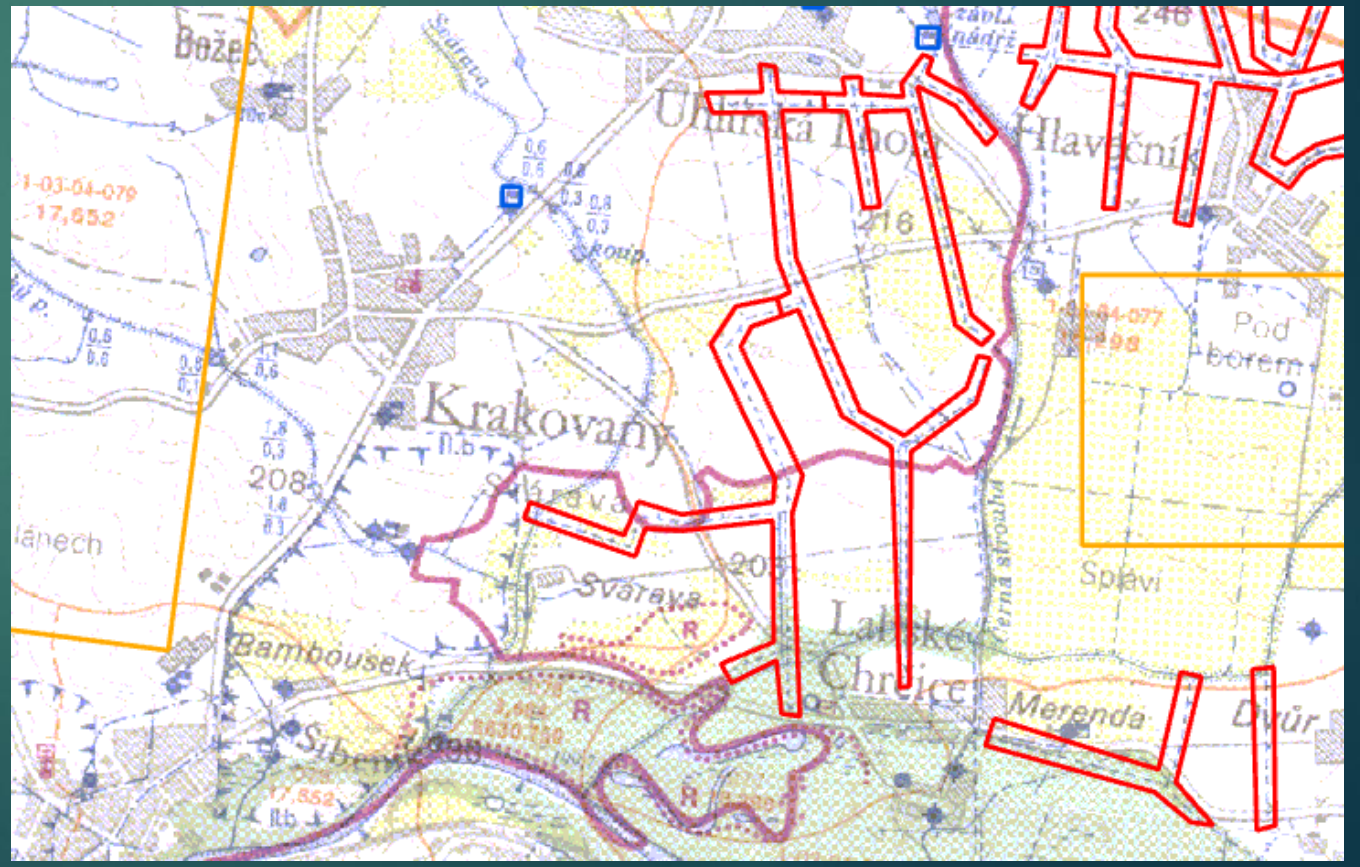
```python
#%%

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(int(labels[i]))
        plt.axis("off")

print("Plotting done")

#%%

##data_augmentation = keras.Sequential(
##    [
##        layers.experimental.preprocessing.RandomFlip("horizontal"),
##        layers.experimental.preprocessing.RandomRotation(0.1),
##    ]
##)

#%%

def make_model(input_shape, num_classes):
    inputs = keras.Input(shape=input_shape)
    # Image augmentation block

##    data_augmentation = keras.Sequential(
##    [
##        layers.experimental.preprocessing.RandomFlip("horizontal"),
##        layers.experimental.preprocessing.RandomRotation(0.1),
##    ]
##    )

##    x = data_augmentation(inputs)
    x = inputs

    # Entry block
    x = layers.experimental.preprocessing.Rescaling(1.0 / 255)(x)
    x = layers.Conv2D(32, 3, strides=2, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    x = layers.Conv2D(64, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    previous_block_activation = x  # Set aside residual

    for size in [128, 256, 512, 728]:
        x = layers.Activation("relu")(x)
        x = layers.SeparableConv2D(size, 3, padding="same")(x)
        x = layers.BatchNormalization()(x)

        x = layers.Activation("relu")(x)
        x = layers.SeparableConv2D(size, 3, padding="same")(x)
        x = layers.BatchNormalization()(x)

        x = layers.MaxPooling2D(3, strides=2, padding="same")(x)

        # Project residual
        residual = layers.Conv2D(size, 1, strides=2, padding="same")(
```

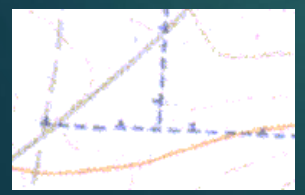
```python
    previous_block_activation = x  # Set aside residual

    for size in [128, 256, 512, 728]:
        x = layers.Activation("relu")(x)
        x = layers.SeparableConv2D(size, 3, padding="same")(x)
        x = layers.BatchNormalization()(x)

        x = layers.Activation("relu")(x)
        x = layers.SeparableConv2D(size, 3, padding="same")(x)
        x = layers.BatchNormalization()(x)

        x = layers.MaxPooling2D(3, strides=2, padding="same")(x)

        # Project residual
        residual = layers.Conv2D(size, 1, strides=2, padding="same")(
            previous_block_activation
        )
        x = layers.add([x, residual])  # Add back residual
        previous_block_activation = x  # Set aside next residual

    x = layers.SeparableConv2D(1024, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    x = layers.GlobalAveragePooling2D()(x)
    if num_classes == 2:
        activation = "sigmoid"
        units = 1
    else:
        activation = "softmax"
        units = num_classes

    x = layers.Dropout(0.5)(x)
    outputs = layers.Dense(units, activation=activation)(x)
    return keras.Model(inputs, outputs)

##image_size = (180, 180)

topgis_model = make_model(input_shape=image_size + (3,), num_classes=2)
##keras.utils.plot_model(topgis_model, show_shapes=True)

epochs = 20

callbacks = [keras.callbacks.ModelCheckpoint("save_at_{epoch}.h5"),]

topgis_model.compile(optimizer=keras.optimizers.Adam(1e-3), loss="binary_crossentropy", metrics=["accuracy"],)

topgis_model.fit( train_ds, epochs=epochs, callbacks=callbacks, validation_data=val_ds,)

topgis_model.save("S:/Private/_PROJEKTY/2020_TACR_DPZ/mapa_udalosti/Tejkl_reseni/topgis_model_2")

def analyse_mosaic(mosaic, image_size, input_model):
    img = keras.preprocessing.image.load_img(mosaic, target_size=image_size)
    img_array = keras.preprocessing.image.img_to_array(img)
    img_array = tf.expand_dims(img_array, 0)  # Create batch axis
    predictions = input_model.predict(img_array)
    score = predictions[0]
    print(    "This image is %.2f percent NoRill and %.2f percent Rill."    % (100 * (1 - score), 100 * score))
```

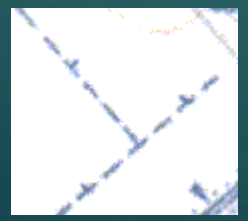
Fakulta agrobiologie,
potravinových a přírodních zdrojů

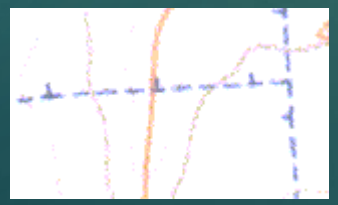
Skeny Základních Vodohospodářských Map

- Konec aktualizací v roce 1997
- Měřítko1:50 000
- Rozlišení 3,18 m, 255 barev
- 220 mapových listů pro ČR
- Veškeré důležité vodohospodářské prvky



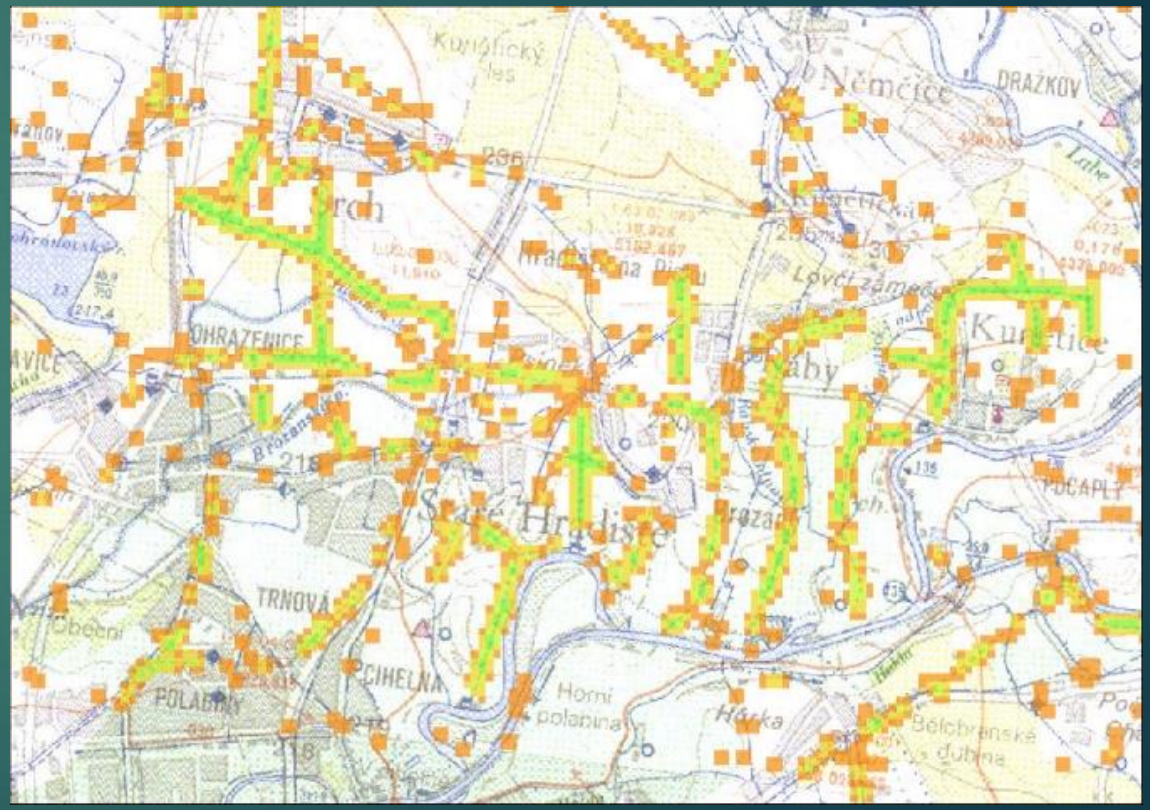






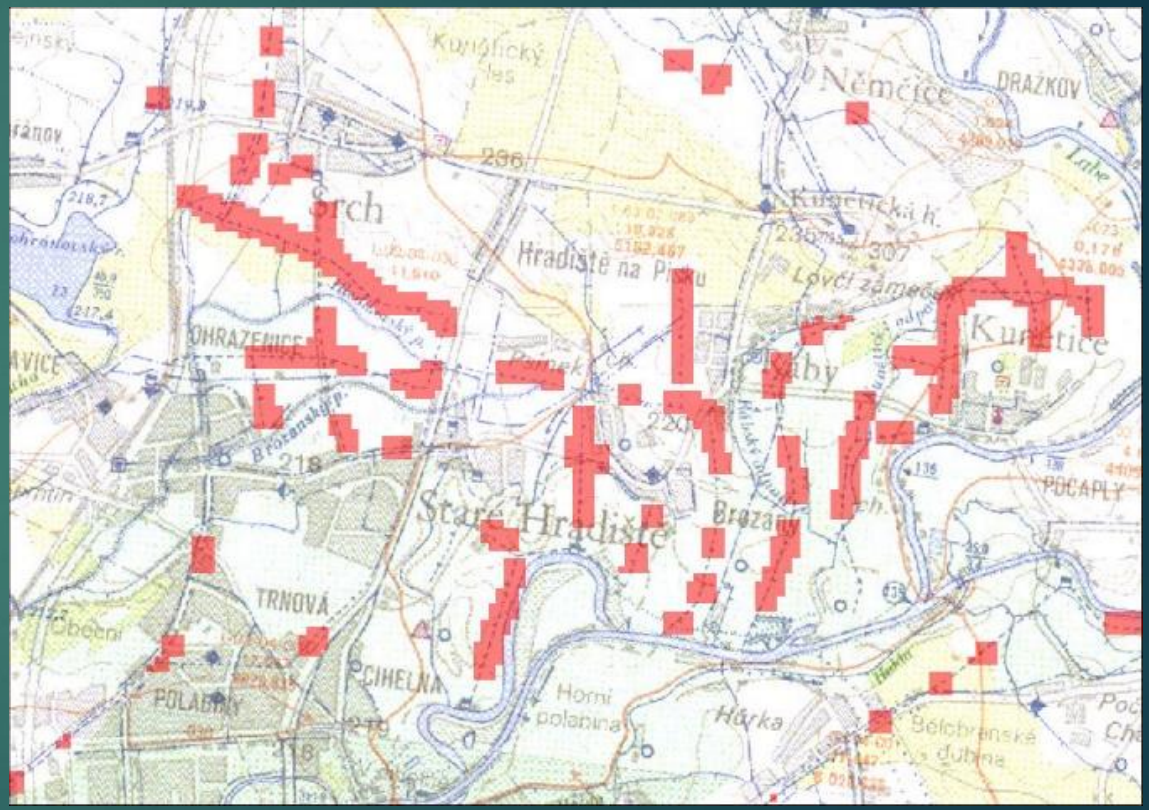
Výsledek klasifikace

- Segmenty s pravděpodobností výskytu jednotlivých tříd 0 – 100 %
- Mnoho osamocených ploch
- Chyby
  - Vodní přivaděč Želivka
  - Kanalizační sběrače
- Ruční kontrola



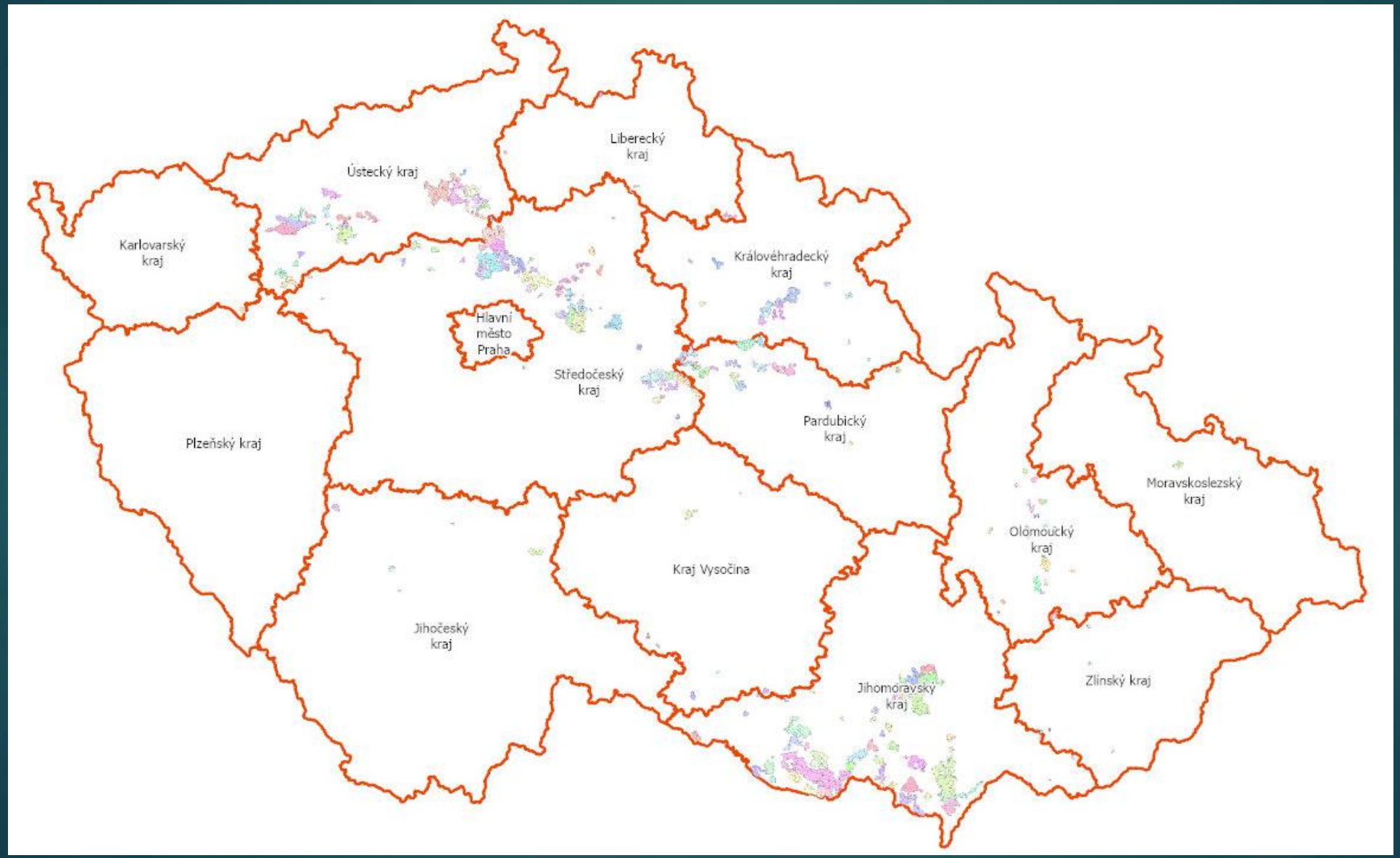
Průměrování, filtrace a převod na polygony

- Klasifikované posunuté listy zprůměrovány
- Segmenty s pravděpodobností menší 50% odstraněny
- Osamocené segmenty odstraněny
- Segmenty s pravděpodobností větší 75% a jejich okolí ponechány
- Ruční kontrola polygonů



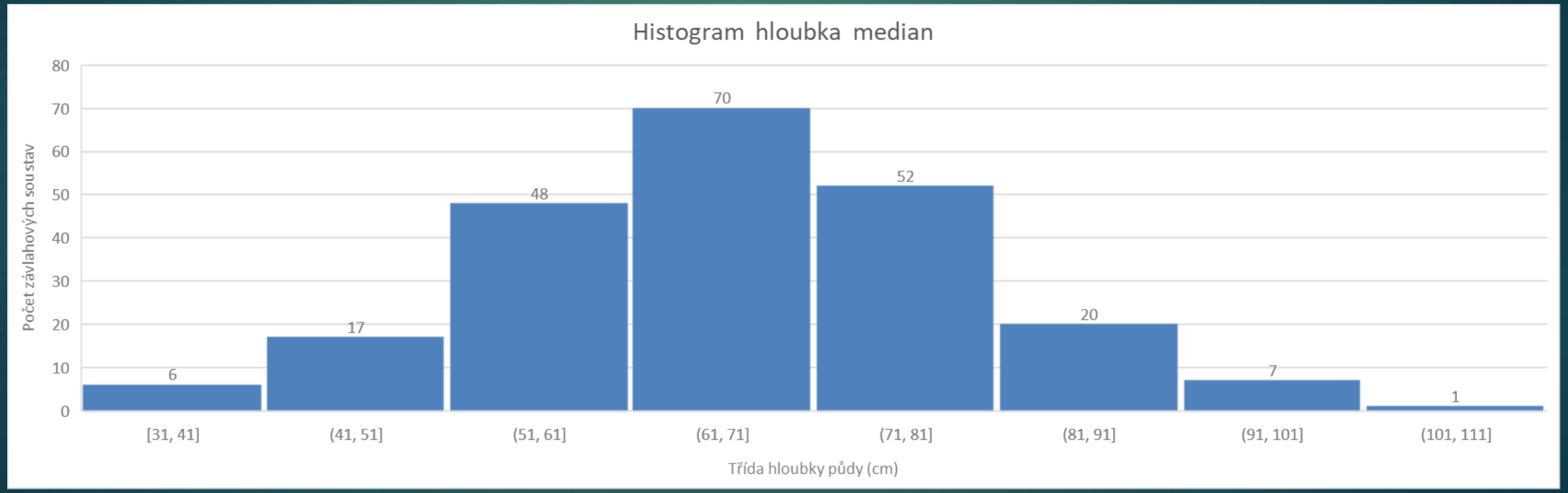
Fakulta agrobiologie,
potravinových a přírodních zdrojů

Vybrané LPIS půdní bloky



Plocha závlahových soustav dle toků

Závlahové soustavy dle půdních vlastností



Fakulta agrobiologie,
potravinových a přírodních zdrojů

Podpora

- Technologická agentura ČR (výzkumný projekt TH02030428 a SS01020052)

Fakulta agrobiologie,
potravinových a přírodních zdrojů

Děkuji za pozornost

Ing. Adam Tejkl Ph.D.

tejkl@af.czu.cz