DEPARTMENT OF LANDSCAPE WATER CONSERVATION

EGU General Assembly 2022

Identification of erosion rills via machine learning

ING. ADAM TEJKL

ING. PETR KAVKA, PH.D.

CTU IN PRAGUE

FACULTY OF CIVIL ENGINEERING

THE DEPARTMENT OF LANDSCAPE WATER CONSERVATION

ADAM.TEJKL@FSV.CVUT.CZ

DEPARTMENT OF LANDSCAPE
WATER CONSERVATION

EGU General Assembly 2022

Motivation

- The laborious and slow process of manual digitization of rills
- Existing large database of erosion damage
- Software and computing capacity is available

Goals

- To create a methodology for Assembly and calibrating of the model
- Minimum requirements for time and performance of the operator

High resolution photogrammetry

EGU General Assembly 2022

- High-resolution soil surface images created using the Structure from Motion photogrammetric method
- Complete with multi-view stereo (MVS)This method is chosen because of the balance between data quality, data acquisition speed and equipment costs
- The photo scan contains many shots of the surface at different angles and several vertical stereo photos
- The distance between the camera and the sample is several tenths of centimeters
- Experimental vessel equipped with reference points and LED lighting
- The photos are merged to create a dense cloud of dots containing up to millions of dots from a cloud of points directly in the software interpolated digital elevation models (DEM) and orthophoto of the surface

DEPARTMENT OF LANDSCAPE
WATER CONSERVATION

EGU General Assembly 2022

Technology

- Hand digitizing of rills on orthophoto scene
- Creation of multiband composite
- Creation of composite mosaic and its export
- Modification of training data
- Model training
- Using a trained model
- Import of results back into GIS

DEPARTMENT OF LANDSCAPE
WATER CONSERVATION

EGU General Assembly 2022

Creation of mosaics from a multiband raster and their export

DEPARTMENT OF LANDSCAPE
WATER CONSERVATION

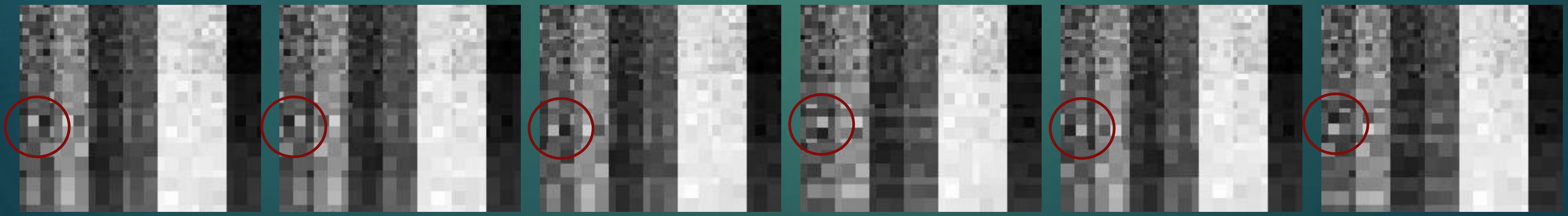
EGU General Assembly 2022

Mosaics

- Variable size of classification window
- R, G, B bands
- Grayscale, Canny Edge
- Grayscale = (0,3 * Red) + (0,59 * Green) + (0,11 * Blue) – linear approximation
- Standardization [0, 1]
- Export as jpg format

Modification of training data

- Only for rill data
- Horizontal and vertical flip
- 90°, 180°, 270° turn
- In order to increase the number of training data



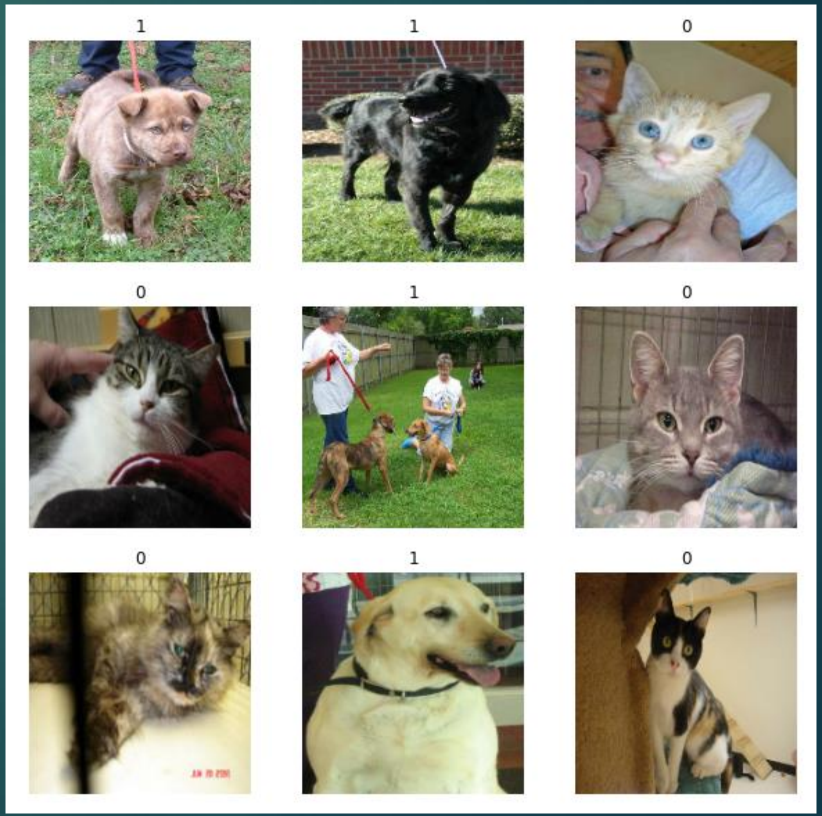
DEPARTMENT OF LANDSCAPE
WATER CONSERVATION

EGU General Assembly 2022

Model

- Python, Tensorflow, Keras
- Deep learning API
- Flexible, Easy, Efective
- Used by NASA, YouTube, Waymo (Chollet 2021)

- Kaggle Cats vs Dogs used as a foundation
- 23 410 training pictures of cats and dogs
- Binary classification



DEPARTMENT OF LANDSCAPE
WATER CONSERVATION

Model

EGU General Assembly 2022

```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))
```

DEPARTMENT OF LANDSCAPE
WATER CONSERVATION

EGU General Assembly 2022

Testing on the data

- Laboratory rainfall simulator
- Experimental throughs in Jirkov
- Long term erosion research

DEPARTMENT OF LANDSCAPE WATER CONSERVATION

EGU General Assembly 2022

Mosaic model for classification

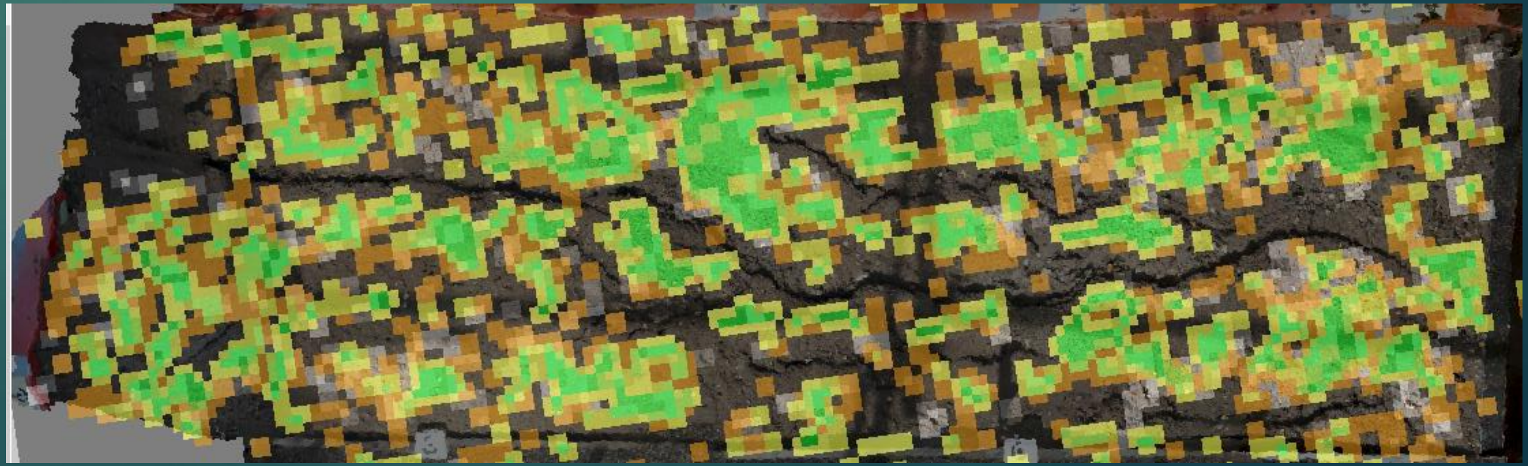
- RGB orthophotos of surface
- Manually digitalized rills
- Category sheet erosion added
- In total 32 orthophotos of surface used
- Mosaics R, G, B, Grayscale, Canny edge 1, Canny edge 2, Canny edge 3

DEPARTMENT OF LANDSCAPE
WATER CONSERVATION

EGU General Assembly 2022

Adding of sheet erosion category

- 27 844 mosaics of no erosion class
- 168 570 mosaics of rill erosion class
- 15 188 mosaics of sheet erosion class
- Validated on 14 orthophotos



DEPARTMENT OF LANDSCAPE WATER CONSERVATION

EGU General Assembly 2022

Testing and tuning of superparameters

- The size of the classification window is a key parameter
- Minimum window size - maximum classification accuracy
- Gradual testing of window sizes and statistical evaluation of these tests
- Another parameter of the model is the spatial resolution of the orthophoto image
- Comparison of the pixel edge length of the image and individual surface elements
- Monitoring the relationship between image resolution and the size of the classification window

DEPARTMENT OF LANDSCAPE WATER CONSERVATION

EGU General Assembly 2022

Future work

- Methodology for success measurement
- Testing and tuning of superparameters
- Searching for the minimal number of training data
- Statistical analysis of results

Funding

- Technological Agency of the Czech Republic (research project TH02030428)
- Internal student CTU grant (SGS17/173/OHK1)

DEPARTMENT OF LANDSCAPE WATER CONSERVATION

EGU General Assembly 2022

Thank you for your attention

Ing. Adam Tejkl

adam.tejkl@fsv.cvut.cz