Web Service for Remote Sensing Scene Classification (RS2C) User Manual

Background

The number of remote sensing platforms for monitoring the surface of the Earth has been growing at ever increasing pace. Consequently, the amount of remote sensing images has surpassed our abilities for manual analysis. Luckily, recent advances in computer vision have provided us with new tools for automatic analysis of remote sensing images.

Scene classification is a key step in remote sensing image analysis aiming to annotate each image with labels from a pre-defined set. These annotations can be used for applications like land cover and land use classification, monitoring urban growth, monitoring and forecasting climate changes, to name only a few. Recently, using remote sensing for monitoring of ecosystems, insects and animals also gains in significance.

Convolutional neural networks (CNNs) have become a de facto standard in various computer vision tasks, ranging from image classification to object detection to semantic segmentation. However, training of CNNs requires considerable computing resources as well as large labeled training sets. In order to lift the burden of CNN training and enable end-users to reap the benefits of using powerful CNN-based classifiers, RS2C offers access to pre-trained CNN models for remote sensing scene classification. In other words, RS2C aims to assist in making sense of the remote sensing imagery acquired using various sensing platforms, ranging from satellites to UAVs. The default problem in remote sensing scene classification is single-label multi-class classification. However, in many cases remote sensing images cannot be accurately described using a single label. For that reason, RS2C also features multi-label classification. Main user communities envisaged to benefit from this service are in the areas of agriculture, food production, urban planning, and environment protection, but practitioners in other fields may also find it useful.

Service Description

RS2C is a RESTful web service and web application for remote sensing scene classification based on convolutional neural networks. Currently, ResNet-50 [1] pre-trained on ImageNet and fine-tuned on MLRSNet [2] is used for classification. The web service is implemented in Python using TensorFlow Serving and Flask. The RS2C API provides two groups of methods:

- Single-label classification - each image is classified into one of the following 46 mutually exclusive scene categories: airplane, airport, bareland, baseball diamond, basketball court, beach, bridge, chaparral, cloud, commercial area, dense residential area, desert, eroded farmland, farmland, forest, freeway, golf course, ground track field, harbor & port, industrial area, intersection, island, lake, meadow, mobile home park, mountain, overpass, park, parking lot, parkway, railway, railway station, river, roundabout, shipping yard, snowberg, sparse residential area, stadium, storage tank, swimmimg pool, tennis court, terrace, transmission tower, vegetable greenhouse, wetland, wind turbine.
    - `url-api`: POST method accepting a comma-separated list of JPEG image URLs.
    - `upload-api`: POST method accepting the list of JPEG images.

Both methods return a JSON object with a list of categories and corresponding confidences for each image.

- Multi-label classification (tagging) - each image is assigned multiple labels from the list: airplane, freeway, roundabout,,airport,golf course, runway, bare soil, grass, sand, baseball diamond, greenhouse, sea, basketball court, gully, ships, beach, habor, snow, bridge, intersection, snowberg, buildings, island, sparse residential area, cars, lake, stadium, chaparral, mobile home, swimming pool, cloud, mountain, tanks, containers, overpass, tennis court, crosswalk, park, terrace, dense residential area, parking lot, track, desert, parkway, trail, dock, pavement, transmission tower, factory, railway, trees, field, railway station, water, football field, river, wetland, forest, road, wind turbine.
    - `multilabel-url-api`: POST method accepting a comma-separated list of JPEG image URLs.
    - `multilabel-upload-api`: POST method accepting the list of JPEG images.

Both methods return a JSON object with a list of labels and confidences that each label can be assigned to the image.

Examples

To perform multi-class classification of images mediumresidential_58.jpg and bridge_22.jpg we can use the following command line:

```
curl -d "urls=https://raw.githubusercontent.com/risojevicv/NI4OS-RSSC/main/webapp/app/static/images/mediumresidential_58.jpg,https://raw.githubusercontent.com/risojevicv/NI4OS-RSSC/main/webapp/app/static/images/bridge_22.jpg" http://rs2c.etfbl.net/url-api
```

To perform multi-label classification of the same images, we can use:

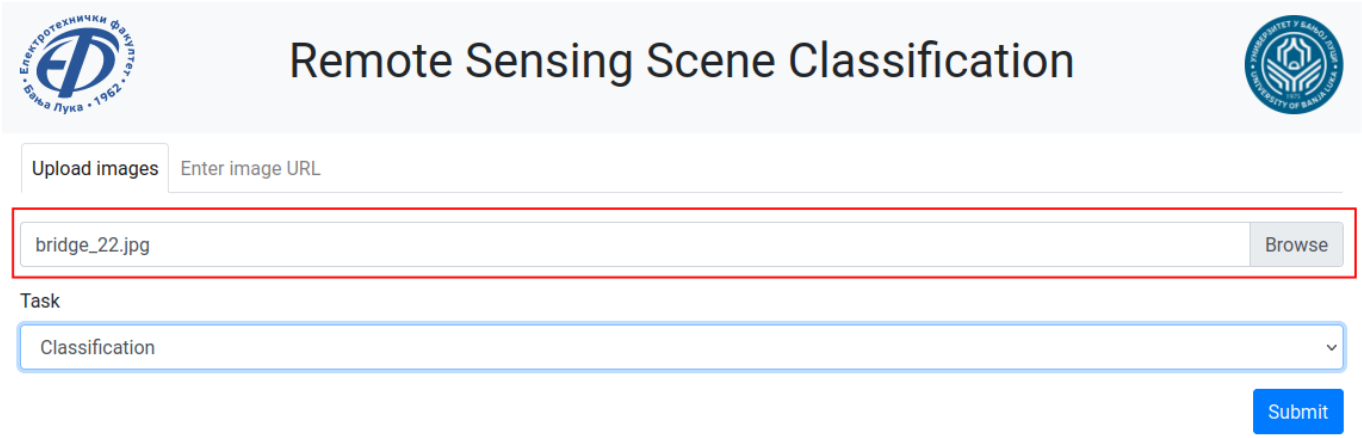
```
curl -d "urls=https://raw.githubusercontent.com/risojevicv/NI4OS-RSSC/main/webapp/app/static/images/mediumresidential_58.jpg,https://raw.githubusercontent.com/risojevicv/NI4OS-RSSC/main/webapp/app/static/images/bridge_22.jpg" http://rs2c.etfbl.net/multilabel-url-api
```

Alternatively, the service can be consumed programmatically. The examples in Python are given in image_classification.py and image_classification_urls.py.

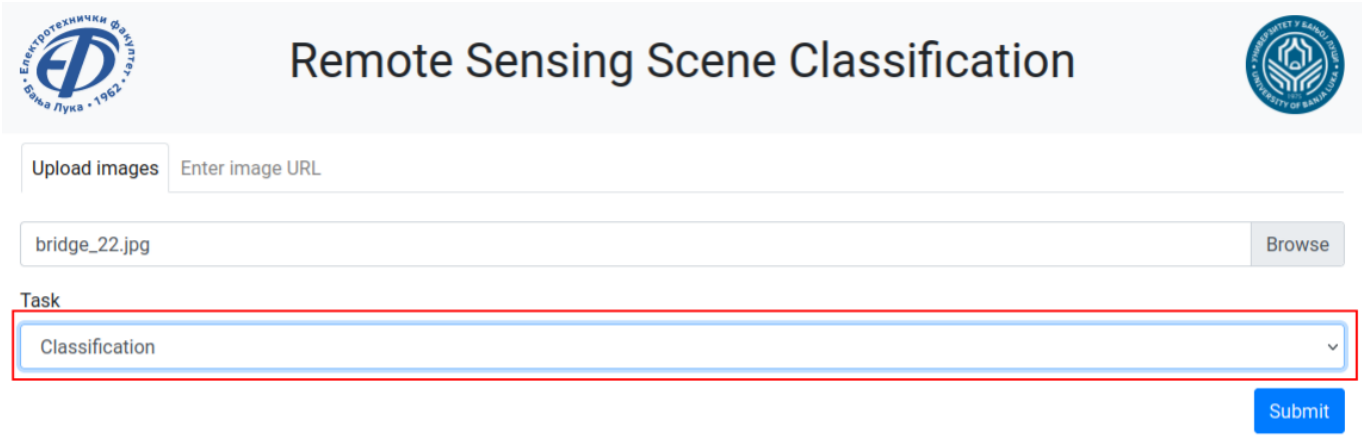
Web application

The service can also be consumed through the web application, available at http://rs2c.etfbl.net. The input images can be uploaded or public URLs provided. In both cases, the user can choose between (single-label) classification and tagging (multi-label classification).

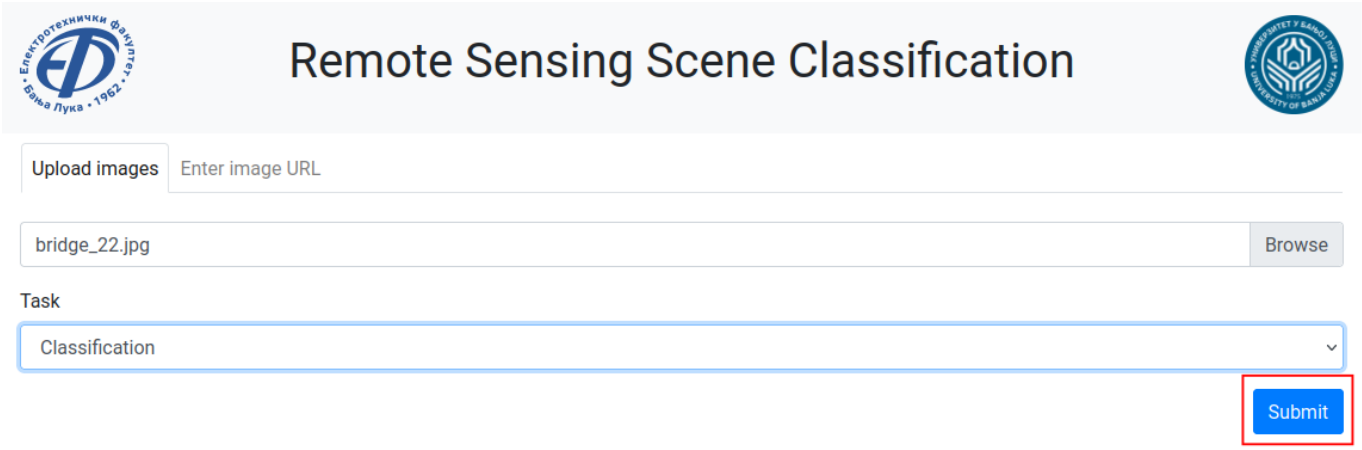
In the case of image upload users should select the locally available images



select the classification or tagging task

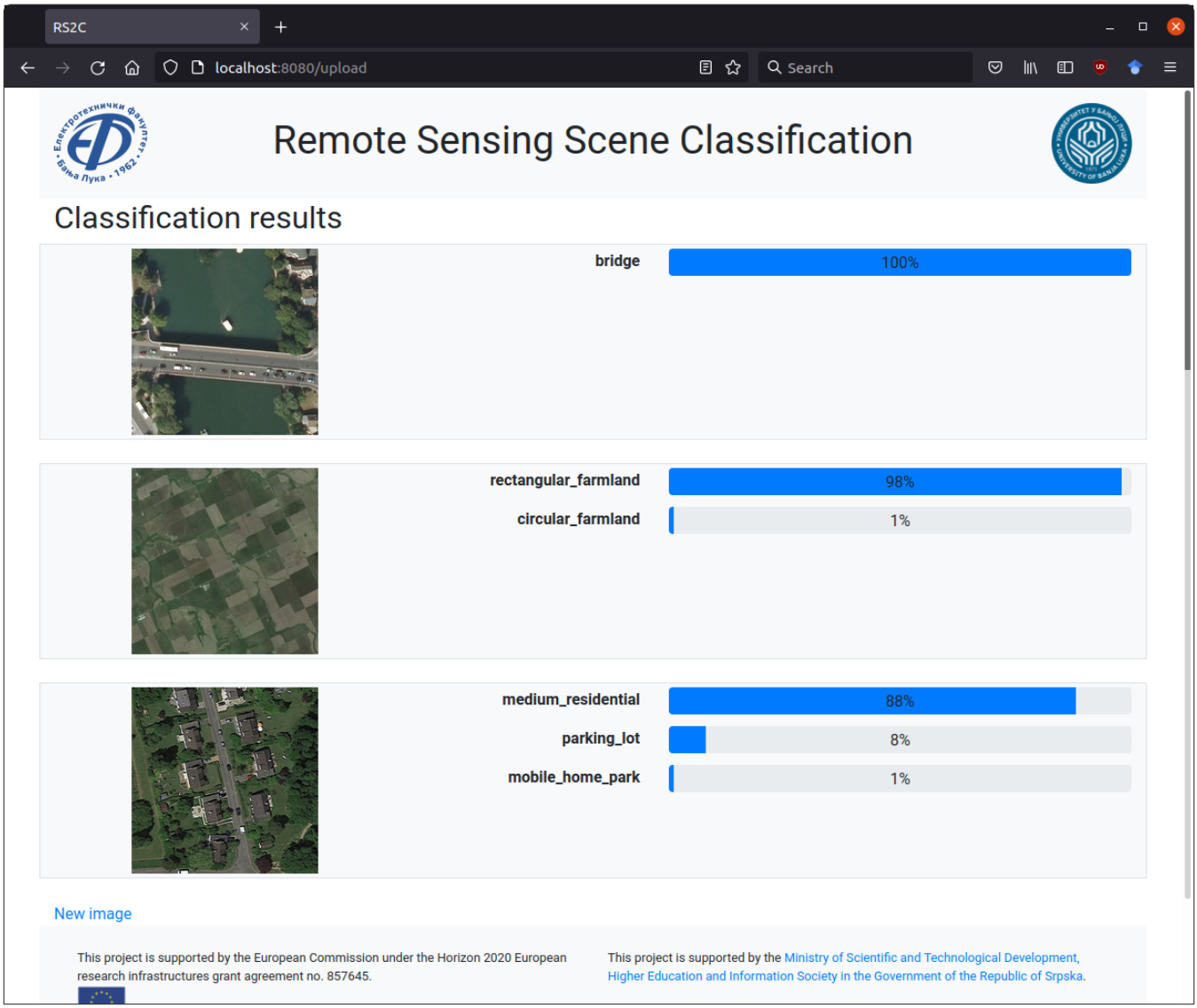


and submit it to RS2C service

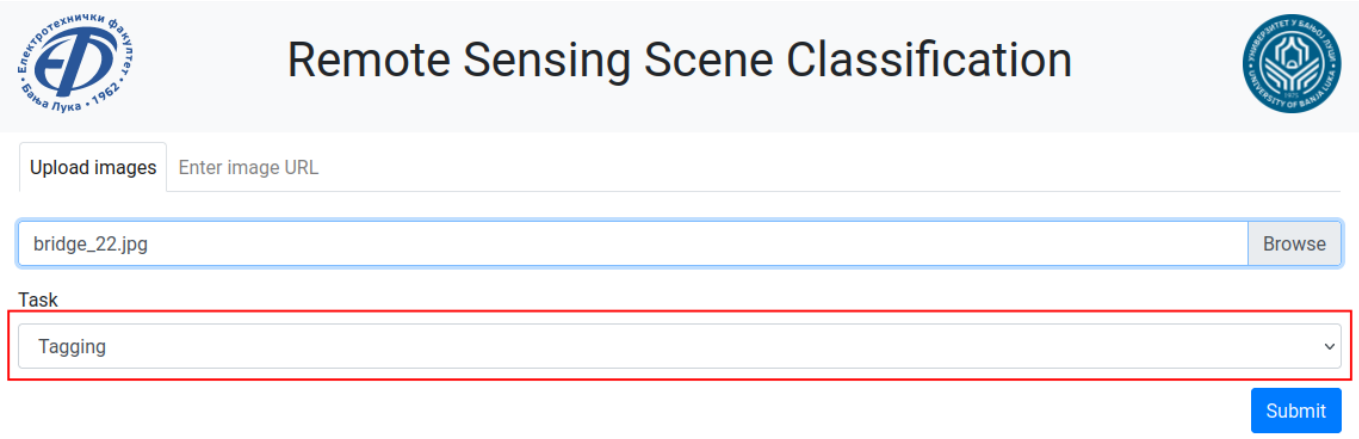


Alternatively, it is possible to select one of the example images

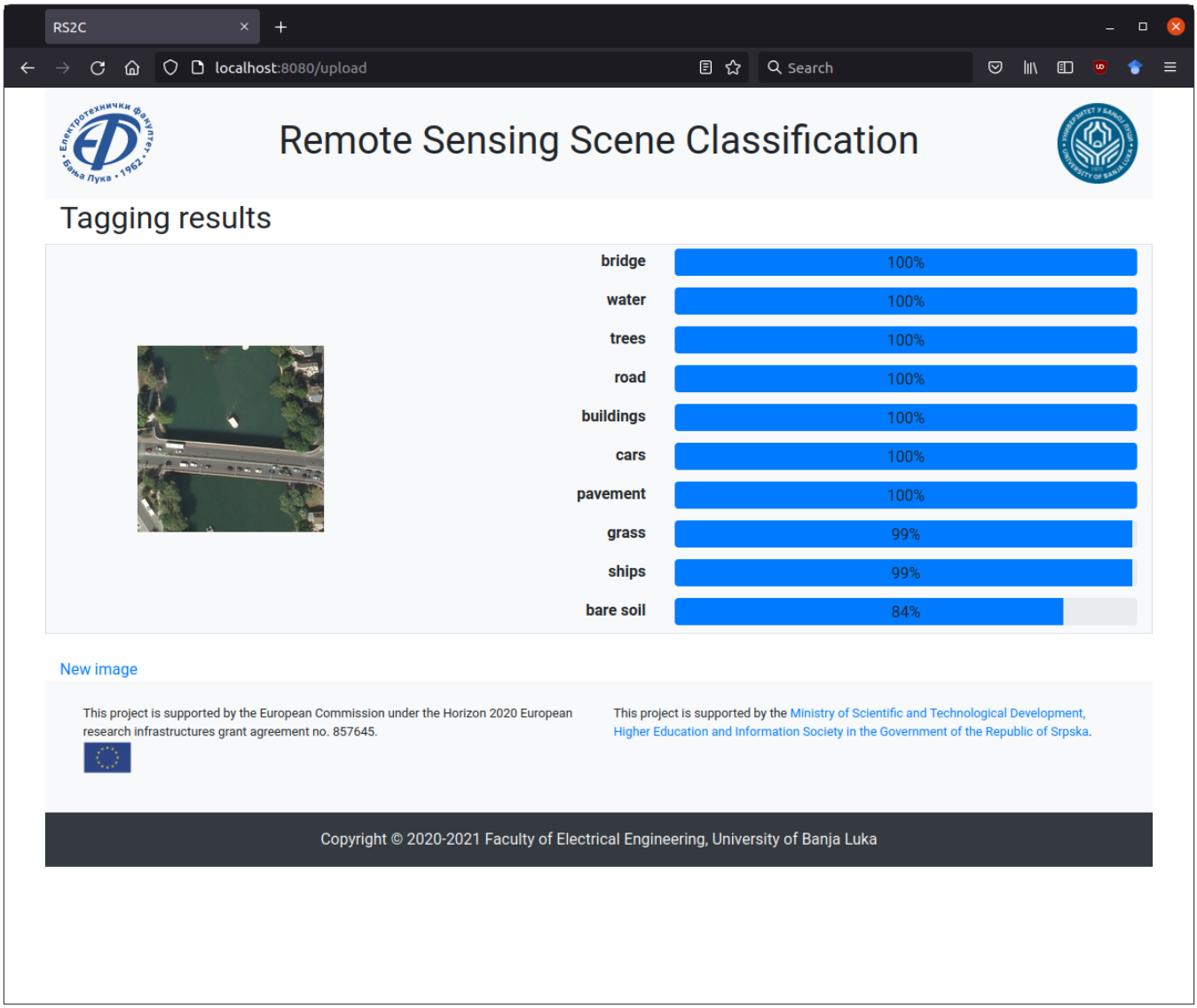
For each uploaded image class predictions along with confidences are given



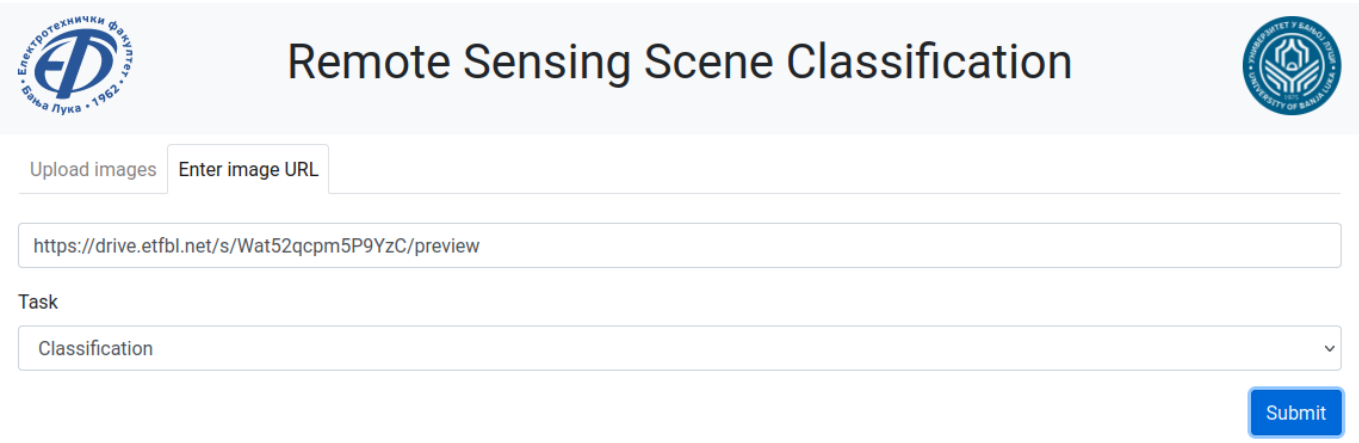
In the case of multi-label classification, tagging should be selected as a task



For each uploaded image predicted labels are given along with confidences. Note that in this case there are multiple labels (tags) that can be assigned to an image



Web application also accepts image URLs for classification or tagging



References

[1] He, Kaiming, et al. "Deep residual learning for image recognition", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[2] Qi, Xiaoman, et al. "MLRSNet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding.", ISPRS Journal of Photogrammetry and Remote Sensing 169 (2020): 337-350.