

# Service description

## Current functionality

The service is intended for image and video file processing. Its NSFW version is aimed at the needs of adult webmasters, tube site owners and cjtubes, where you need the best possible "clickable" images.

The service is in beta and not all functionality is currently available.

### Functionality for screenshots (full-size frames from the video) and their smaller versions (thumbs)

- Rank the images according to their aesthetic appeal. This version does not take into account the category of the images. The algorithm is based on neural networks. There is a separate version adapted for nsfw images. For each image a rating from 0 to 1 is returned.
- Jpeg compression artifact removal and image resolution increase
- *[coming soon]* Image ranking within a certain category.
- *[coming soon]* Recognizing clip categories based on a set of screenshots from that clip
- *[beta]* Finding a logo and its coordinates on a set of images, useful for cropping screenshots without a logo
- *[beta]* Logo database detection - determining which site the video belongs to
- *[beta]* Identification of actors starring in the clip - by database.
- *[beta]* Finding blacklisted objects - animals and children in the image.

### A basic algorithm for working with the image ranking functionality:

- Get screenshots from the clip with twice the margin.
- Send the urls of these screenshots to our API.
- Get a list of screenshots and their rating from 0 to 1 (0 - bad, 1 - good)
- Leave only screenshots with the best rating

**Attention** - we recommend comparing screenshots within one clip and leave the best ones.

## Neuroscore.ai API instructions

The following program Insomnia <https://insomnia.rest/download/core> (Win/Mac/Unix) is handy for debugging and testing

First, make a POST request similar to this one:

- In the header - content-type and received x-api-key
- In the message body - json with the request

```
curl --request POST \  
--url https://api.neuroscore.ai/api/v1/tasks \  
--header 'content-type: application/json' \  
--header 'x-api-key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx' \  
--data '{  
  "urls": [  
    "https://domain.com/image1.jpg",  
    "https://domain.com/image2.jpg",  
    "https://domain.com/image3.jpg",  
    "https://domain.com/image4.jpg"  
  ],  
  "tasks": [  
    "score"  
  ]  
}'
```

You'll get "202 Accepted" and the following json in the response:

```
{  
  "apiVersion": "1.0",  
  "data": {  
    "href": "http://api.neuroscore.ai/api/v1/tasks/04c825cd-2afc-523c-afe9-fb5a733a46af",  
    "task_id": "04c825cd-2afc-523c-afe9-fb5a733a46af",  
    "is_new": false,  
    "balance_points": 49670  
  }  
}
```

Make a GET request (don't forget the required keys in the request header) for the received url (in our example: <http://api.neuroscore.ai/api/v1/tasks/04c825cd-2afc-523c-afe9-fb5a733a46af>)

We'll receive, depending on the status of the task, either a half-empty or the following answer:

```
{
  "apiVersion": "1.0",
  "data": {
    "href": "https://api.neuroscore.ai/api/v1/tasks/967a335e-327c-55b7-8572-add9a28cbe47",
    "task_id": "967a335e-327c-55b7-8572-add9a28cbe47",
    "time_created": "2020-04-07 14:23:05.860527",
    "time_updated": "2020-04-07 19:03:32.440586",
    "query": {
      "urls": [
        "https://domain.com/image1.jpg",
        "https://domain.com/image2.jpg",
        "https://domain.com/image3.jpg",
        "https://domain.com/image4.jpg"
      ],
      "tasks": [
        "score"
      ]
    },
    "response": [
      {
        "url": "https://domain.com/image1.jpg",
        "info": {
          "width": 640,
          "height": 480,
          "filesize": 112982
        },
        "detection": {
          "score": 0.9464230686
        }
      },
      {
        "url": "https://domain.com/image2.jpg",
        "info": {
          "width": 640,
          "height": 480,
          "filesize": 94209
        },
        "detection": {
          "score": 0.9103920933
        }
      },
      {
        "url": "https://domain.com/image3.jpg",
        "info": {
          "width": 640,
          "height": 480,
          "filesize": 31109
        },
        "detection": {
          "score": 0.39191918
        }
      },
      {
        "url": "https://domain.com/image4.jpg",
        "info": {
          "width": 640,
          "height": 480,
          "filesize": 93314
        },
        "detection": {
          "score": 0.9819819989
        }
      }
    ]
  }
}
```

```
}
}
}
```

A list of possible commands:

```
{{api_path}} = https://api.neuroscore.ai/api/v1
```

- GET requests - to get data
- POST requests - to send data. For now it is only used to create a task

GET {{api\_path}}/tasks?filter=new - a list of unprocessed tasks.

Returns the following data: ``{"apiVersion": "1.0", "data": {"response": ["task-id1", "task-id2", ... "task-id3"]}}``

```
---
```GET {{api_path}}/tasks?filter=done``` - list of tasks with results

Similar response

---

```GET {{api_path}}/tasks/{{task-id1}}``` - get the task data. It may return 202 code if the task is not ready, or 200 and a json wi

---

```GET {{api_path}}/balance``` - get account balance

---

```POST {{api_path}}/tasks``` - create a new task
* urls - a list of image urls
* task = ["score"] - to get image scores
```

```
{"urls": ["https://domain.com/image1.jpg", "https://domain.com/image2.jpg", "https://domain.com/image3.jpg", "https://domain.com/image4.jpg"], "tasks": ["score"]}
}```
```

In Insomnia, you can export the current query in format of any programming language - CTRL+SHIFT+G - and select the output language among php/python/go/shell

## Tips & Tricks

### What is the best way to integrate ranking?

1. Generate twice as many screenshots from the clip as needed. If you need 20 screenshots, generate 40 screenshots.
2. Create a task with screenshot urls within one clip. Save the task id in your database
3. Every few minutes/hours make a GET request /tasks?filter=done, see if the task appears there. If it appeared, get the data through /tasks/{{task-id1}}
4. image rating is a number from 0 to 1. 1 - very good. 0 - very bad. We recommend to keep the images with the highest score, and to delete the ones with the lowest score.

Last Modified on 11/18/2021 13:16:14