Automated Content Moderation

Eduardo Freitas
DATA CAPTURE SPECIALIST

Overview

What Is Automated Image Moderation?

Evaluating Adult & Racy Content

Detecting Faces

Detecting Text with Optical Character Recognition

Profanity Content Checking

Personally Identifiable Information, Auto-correction & Classification

Demos – Smart Moderation (.NET SDK)

What Is Automated Image Moderation?

Flag and filter out content from images that might pose a risk, without human intervention.

Automated Image Moderation Aspects

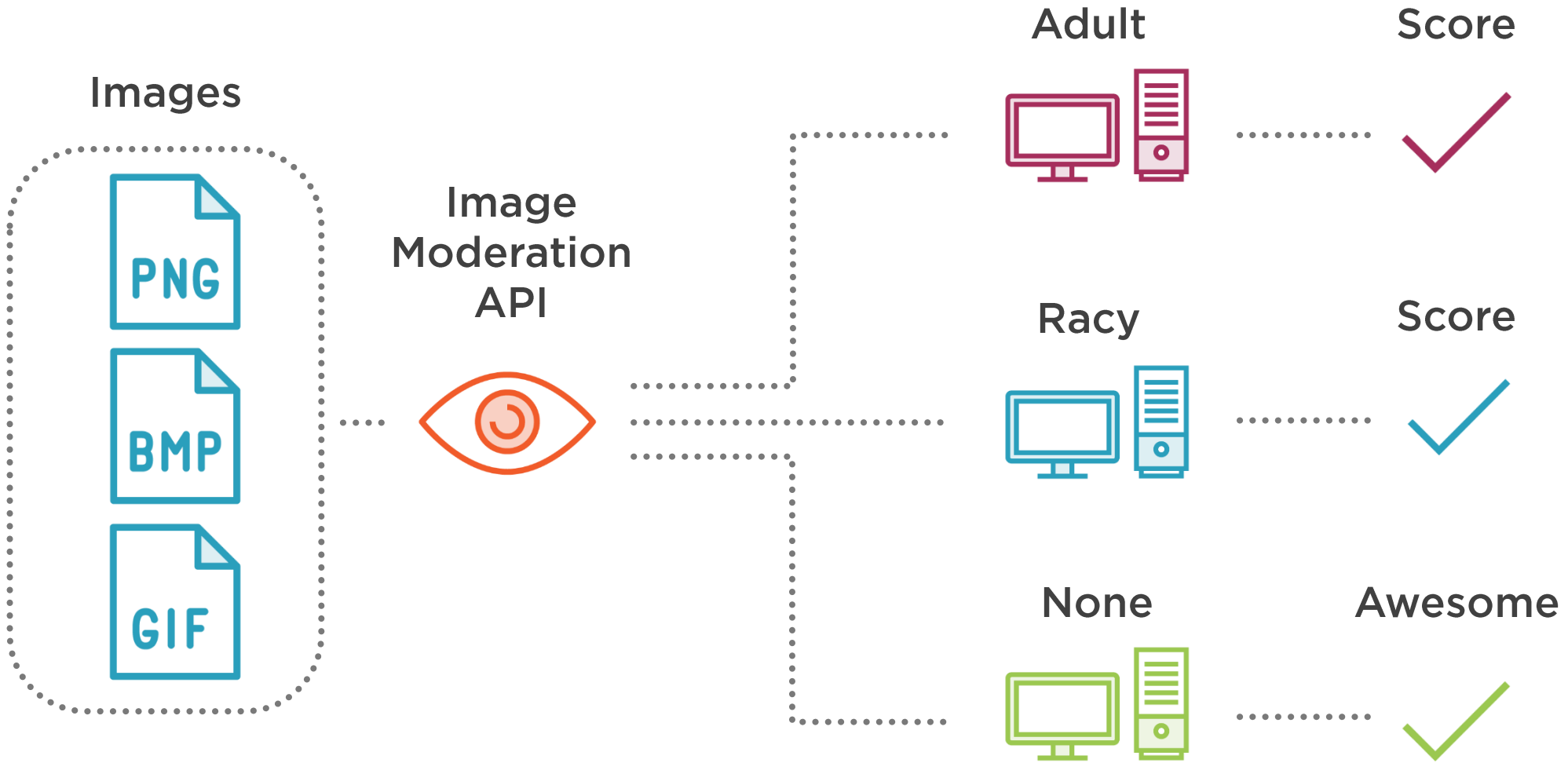
Adult & Racy Content Evaluation

Detecting Faces

Detecting Text with OCR

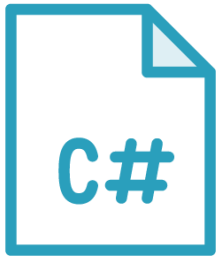
Evaluating Adult & Racy Content

Evaluation Process



What method does the Image Moderation API provide to do this?

Evaluate Image



Microsoft Content Moderator SDK: Windows Client

Available on Github

https://github.com/MicrosoftContentModerator/Microsoft.CognitiveServices.ContentModerator-Windows

Using the .NET SDK - C#

```
public static async Task<EvaluateImageResult> EvaluateImage(string image)
{
    EvaluateImageResult ev = null;
    ModeratorClient client = new ModeratorClient(cSubscriptionKey, cEndpoint);
    if (File.Exists(image))
        ev = await client.EvaluateImageAsync(
            new FileStream(image, FileMode.Open, FileAccess.Read), false);
    return ev;
}
```

Using an SDK saves time!

Detecting Faces

Features

Personally identifiable information (PII)

Detects potential faces in images

Number of potential faces in each image

```
{
"FaceDetection": {
"result": true,
  "count": 1,
  "advancedInfo": [],
  "faces": [{
      "bottom": 598,
      "left": 44,
      "right": 268,
      "top": 374 }]
}
}
```


◀ Indicates if faces are found

◀ Number of faces found

◀ Object of faces found

◀ Coordinates of a face

```
curl -v -X POST
"https://[location].cognitive.microsoft.com/contentmoderato
r/moderate/v1.0/ProcessImage/FindFaces?CacheImage={boolean}
"
-H "Content-Type: application/json"
-H "Ocp-Apim-Subscription-Key: {subscription key}"
--data-ascii "{body}"
```

cURL Faces Detection Example

- location
- Ocp-Apim-Subscription-Key
- CacheImage (optional)
- {body}

Content-Type
image/gif
image/jpeg
image/png
image/bmp
application/json

Raw C# - Using HTTP

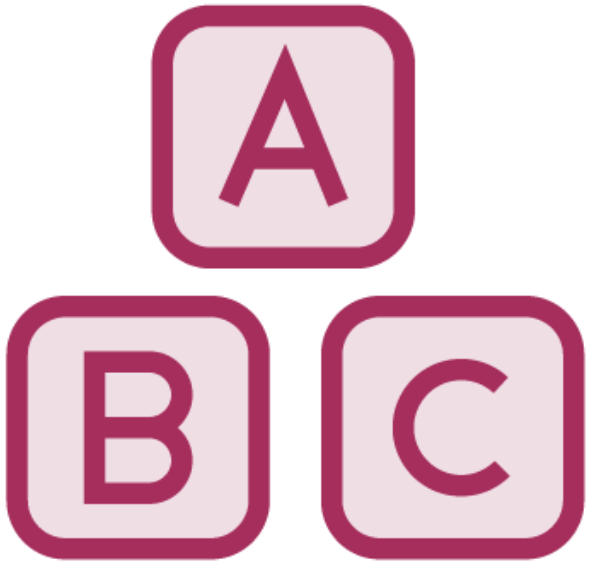
```
public async Task<string> MakeRequest(string image, string contentType, string uri)
{
    string contentString = string.Empty;
    HttpClient client = new HttpClient();
    client.DefaultRequestHeaders.Add(cOcpApimSubscriptionKey, cSubscriptionKey);
    HttpResponseMessage response = null;

    if (File.Exists(image) && uri != string.Empty && contentType != string.Empty)
    {
        byte[] byteData = GetAsByteArray(image);
        using (var content = new ByteArrayContent(byteData))
        {
            content.Headers.ContentType = new MediaTypeHeaderValue(contentType);
            response = await client.PostAsync(uri, content);
            contentString = await response.Content.ReadAsStringAsync();
        }
    }
    return contentString;
}
```

Using the .NET SDK - C#

```
public static async Task<DetectFacesResult> FindFacesImage(string image)
{
    DetectFacesResult df = null;
    ModeratorClient client = new ModeratorClient(cSubscriptionKey, cEndpoint);
    if (File.Exists(image))
        df = await client.DetectFacesImageAsync(
            new FileStream(image, FileMode.Open, FileAccess.Read), false);
    return df;
}
```

Detecting Text with OCR

Features



Ability to extract text from an image

Response includes the original text

Detected text elements and their scores

```
"TextDetection": {
  "status": {
    "code": 3000.0,
    "description": "OK",
    "exception": null
  }, "language": "eng",
  "text": "IF WE DID \r\nALL \r\nTHE
THINGS \r\nWE ARE \r\nCAPABLE \r\nOF
DOING, \r\nWE WOULD \r\nLITERALLY
\r\nASTOUND \r\nOURSELVE \r\n",
  "candidates": []
}
```


◀ Description of the result obtained

◀ Language detected

◀ Text extracted from the image

```
curl -v -X POST
"https://[location].cognitive.microsoft.com/contentmoderato
r/moderate/v1.0/ProcessImage/OCR??language={string}&CacheIm
age={boolean}&enhanced={boolean}"
-H "Content-Type: application/json"
-H "Ocp-Apim-Subscription-Key: {subscription key}"
--data-ascii "{body}"
```

cURL OCR Example

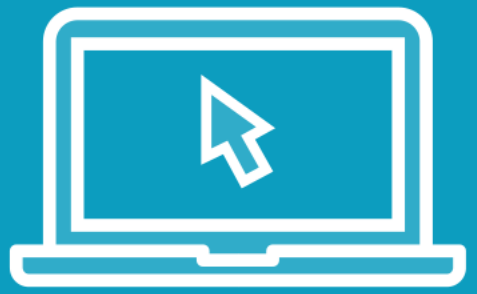
- location
- Ocp-Apim-Subscription-Key
- CacheImage (optional)
- {body}

Content-Type
image/gif
image/jpeg
image/png
image/bmp
application/json

Using the .NET SDK - C#

```
public static async Task<OcrImageResult> OcrImage(string image)
{
    OcrImageResult df = null;
    ModeratorClient client = new ModeratorClient(cSubscriptionKey, cEndpoint);
    if (File.Exists(image))
        df = await client.OCRImageAsync(
            new FileStream(image, FileMode.Open, FileAccess.Read), false);
    return df;
}
```

Demo



Smart image moderation

Evaluate images for racy and adult content

Detect faces within an image

Extract text using OCR

What Is Automated Text Moderation?

Flag and filter out text that might classify as undesirable content, without human intervention.

Automated Text Moderation Aspects

Profanity & Screening

Personally Identifiable Information (PII)

Auto-correction

Text Screening

Features

- Ability to flag profanity terms
- Classifies for possible undesired text
- Personally Identifiable Information (PII)
- Auto-correction

Personally Identifiable Information (PII)

- Email addresses
- US Mailing addresses
- IP addresses
- US Phone numbers
- UK Phone numbers
- Social Security Numbers (SSN)

The qu!ck brown f0x jumps
over the lzay dog.

◀ The original text

The quick brown fox jumps
over the lazy dog.

◀ The text with auto-correction

```
"Terms": [
{
    "Index": 118,
    "OriginalIndex": 118,
    "ListId": 0,
    "Term": "crap"
}]
```


◀ Indicates the location on the text

◀ Profanity term found

```
curl -v -X POST
"https://[location].cognitive.microsoft.com/contentmoderator/moderate/v1.0/ProcessText/Screen?autocorrect={boolean}&PII={boolean}&listId={string}&classify={boolean}&language={string}"

-H "Content-Type: application/json"

-H "Ocp-Apim-Subscription-Key: {subscription key}"

--data-ascii "{body}"
```

cURL Screen Example

- location
- Ocp-Apim-Subscription-Key
- autocorrect (optional)
- {body}

Content-Type
text/html
text/xml
text/markdown
text/plain

Using the .NET SDK - C#

```
public static async Task<ScreenTextResult> ScreenText(string file)
{
    ScreenTextResult tr = null;
    ModeratorClient client = new ModeratorClient(cSubscriptionKey, cEndpoint);
    if (File.Exists(file))
        tr = await client.ScreenTextAsync(File.ReadAllText(file),
            Constants.MediaType.Plain, "eng", true, true, true, string.Empty);
    return tr;
}
```

Demo

Smart text moderation

Detect profanity content

Detect Personally Identifiable Information (PII)

Perform Auto-correction

Summary

- Automated Moderation
- Adult & Racy Content
- Detecting Faces
- Text with Optical Character Recognition
- Profanity Content Checking
- Personally Identifiable Information
- Demos – Smart Moderation (.NET SDK)