Make It Aesthetic

Anna-Maria Hohmann
Konrad Kring
Meret Lindanis

6th presentation
14.01.2020

About Us

Meret

- computer science
- medical technologies

Anna

- business informatics
- project management

Konrad

- pedagogics
- music

Goals of Our Project

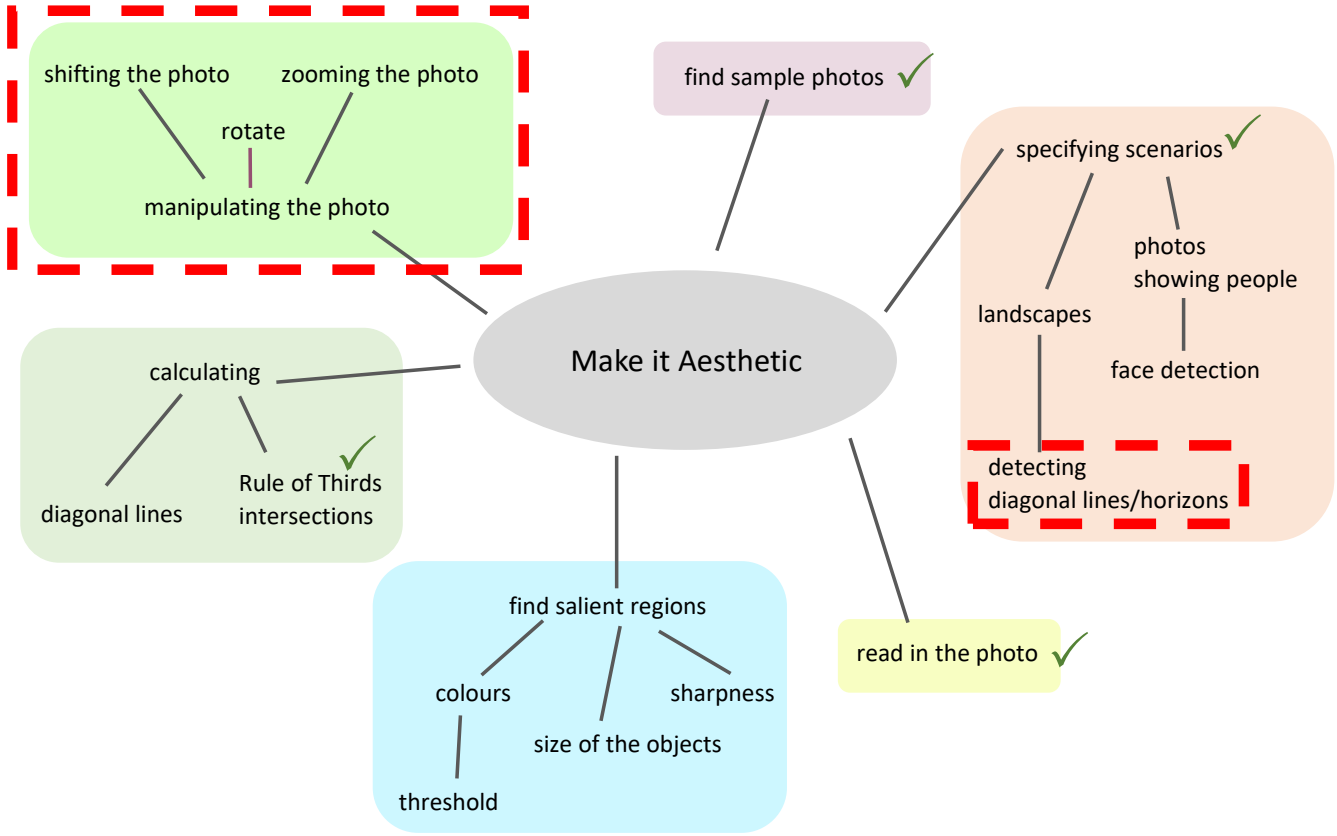
our motivation:

- interested in photography
- opening aesthetic photography to the public
- simplifying the aesthetic photography for the user
- being able to save every moment in beautiful photos
- bringing this knowledge into school

Goals of Our Project

- make given photos aesthetic
- by zooming, rotating or cropping the photo
- selecting the guideline the photo should follow

Milestones


shifting the photo
zooming the photo
rotate
manipulating the photo
find sample photos ✓
specifying scenarios ✓
photos showing people
landscapes
face detection
detecting diagonal lines/horizons
Make it Aesthetic
calculating
diagonal lines
Rule of Thirds intersections ✓
find salient regions
colours
sharpness
size of the objects
threshold
read in the photo ✓

Implementation

Filter of detected lines which are too steep

```
def aggregate_to_horizon(lines):
  horizon = math.inf
  for line in lines:
    line = line[0]
    begin_x = line[0]
    begin_y = line[1]
    end_x = line[2]
    end_y = line[3]

    gradient = (begin_y - end_y) / (begin_x - end_x)

    is_flat_enough = abs(gradient) < 0.1

    if is_flat_enough:
      if begin_y < horizon:
        horizon = begin_y
      if end_y < horizon:
        horizon = end_y

  return horizon
```

Implementation

Filter of detected lines which are too steep

Implementation

Filter of detected lines which are too steep

Implementation

Cropping of an image with respect distance between horizon and upper and lower RT-line and aspect ratio

```
def crop_img(img):
  img = img.copy()
  width, height, third_of_height_1, third_of_height_2, third_of_width_1, third_of_width_2 = generate_image_data(img)
  img, edges, dilation, erosion, image_line, lines_edges, lines, horizon = detect_horizon(img)

  # cv2.line(img, (0, horizon), (width, horizon), (255, 0,0), thickness=10, lineType=cv2.LINE_AA)
  upper_horizon_distance = abs(third_of_height_1 - horizon)
  lower_horizon_distance = abs(third_of_height_2 - horizon)
  closer_to_upper_horizon = upper_horizon_distance < lower_horizon_distance

  aspect_ratio = width/height

  top_cropping = 0
  bottom_cropping = 0
  left_cropping = 0
  right_cropping = 0

  if closer_to_upper_horizon:
    top_cropping = int((3/2) * upper_horizon_distance)
  else:
    bottom_cropping = int((3/2) * lower_horizon_distance)

  # determine the left and right cropping_point of the x-asis, to crop the picture middle-weight
  new_height = height - top_cropping - bottom_cropping
  new_width = int(new_height * aspect_ratio)
  left_cropping = (width - new_width) // 2
  right_cropping = left_cropping

  cropped_img = img[top_cropping:(height - bottom_cropping), left_cropping:(width - right_cropping)]

  return cropped_img

img = load_image('IMG_1334.JPG')
plot_cropped_using_rt(img)
```

Implementation

Cropping of an image with respect distance between horizon and upper and lower RT-line and aspect ratio

```
upper_horizon_distance = abs(third_of_height_1 - horizon)
lower_horizon_distance = abs(third_of_height_2 - horizon)
closer_to_upper_horizon = upper_horizon_distance < lower_horizon_distance

aspect_ratio = width/height

top_cropping = 0
bottom_cropping = 0
left_cropping = 0
right_cropping = 0
```

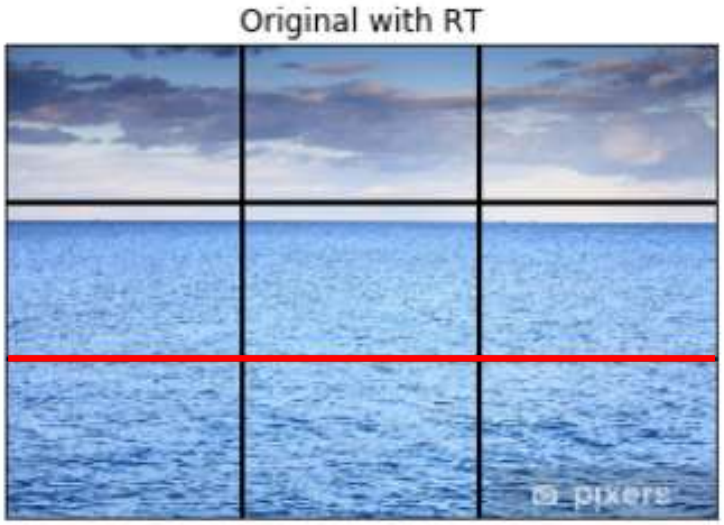
Implementation

Cropping of an image with respect distance between horizon and upper and lower RT-line and aspect ratio

```
if closer_to_upper_horizon:
  top_cropping = int((3/2) * upper_horizon_distance)
else:
  bottom_cropping = int((3/2) * lower_horizon_distance)

# determine the left and right cropping_point of the x-asis, to crop the picture middle-weight
new_height = height - top_cropping - bottom_cropping
new_width = int(new_height * aspect_ratio)
left_cropping = (width - new_width) // 2
right_cropping = left_cropping

cropped_img = img[top_cropping:(height - bottom_cropping), left_cropping:(width - right_cropping)]

return cropped_img
```

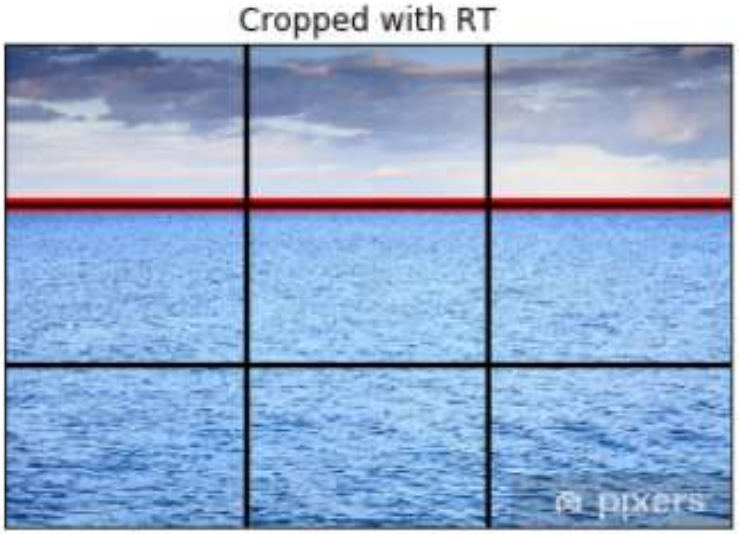
Implementation

Cropping of an image with respect distance between horizon and upper and lower RT-line and aspect ratio



Implementation

Cropping of an image with respect distance between horizon and upper and lower RT-line and aspect ratio



Implementation

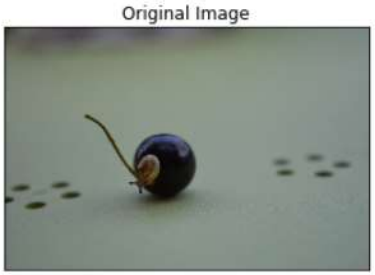
Saliency Detection

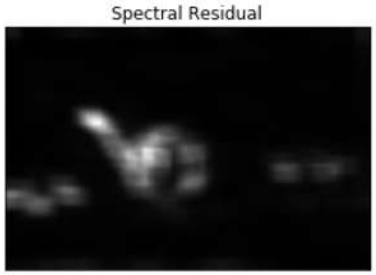
```
def spectralResidualSaliency(img):
  #initialize OpenCV's static saliency spectral residual detector and compute
  #the saliency map
  saliency = cv2.saliency.StaticSaliencySpectralResidual_create()
  (success, saliencyMap) = saliency.computeSaliency(img)

  saliencyMap = (saliencyMap * 255)
  threshMap = cv2.threshold(saliencyMap.astype("uint8"), 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]
  return img, saliencyMap, threshMap
```

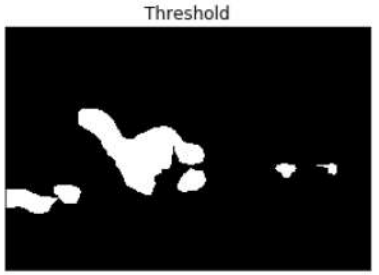
[3]

Implementation

Saliency Detection







Implementation

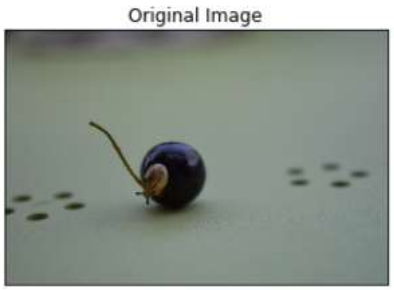
Saliency Detection

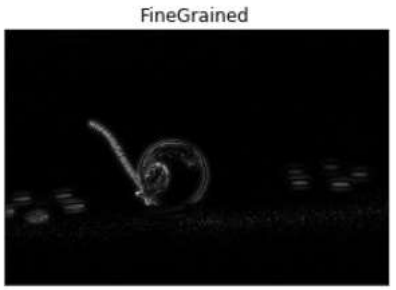
```
def fineGrainedSaliency(img):
  saliencyFG = cv2.saliency.StaticSaliencyFineGrained_create()
  (success, saliencyMap) = saliencyFG.computeSaliency(img)
  #if we would like a *binary* map that we could process for contours,
  #c0ompute convex hull's, extract bounding boxes, etc., we can additionally
  # threshold the saliency map
  threshMap = cv2.threshold(saliencyMap.astype("uint8"), 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]
  return img, saliencyMap, threshMap
```

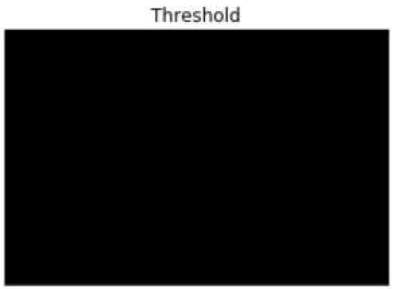
[3]

Implementation

Saliency Detection







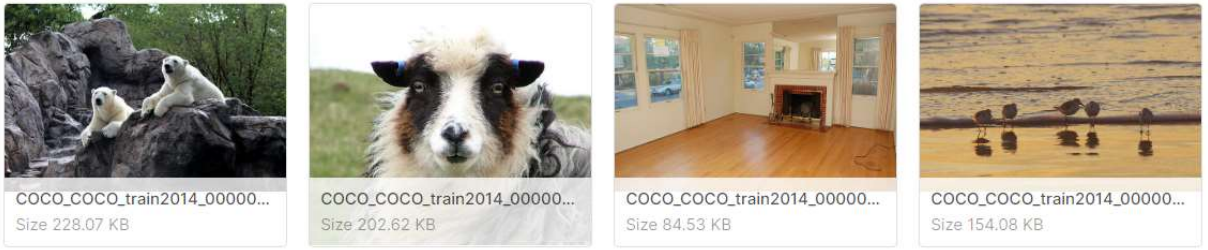
Challenges

- No recognition in threshold for saliency detection in fineGrainedSaliency function

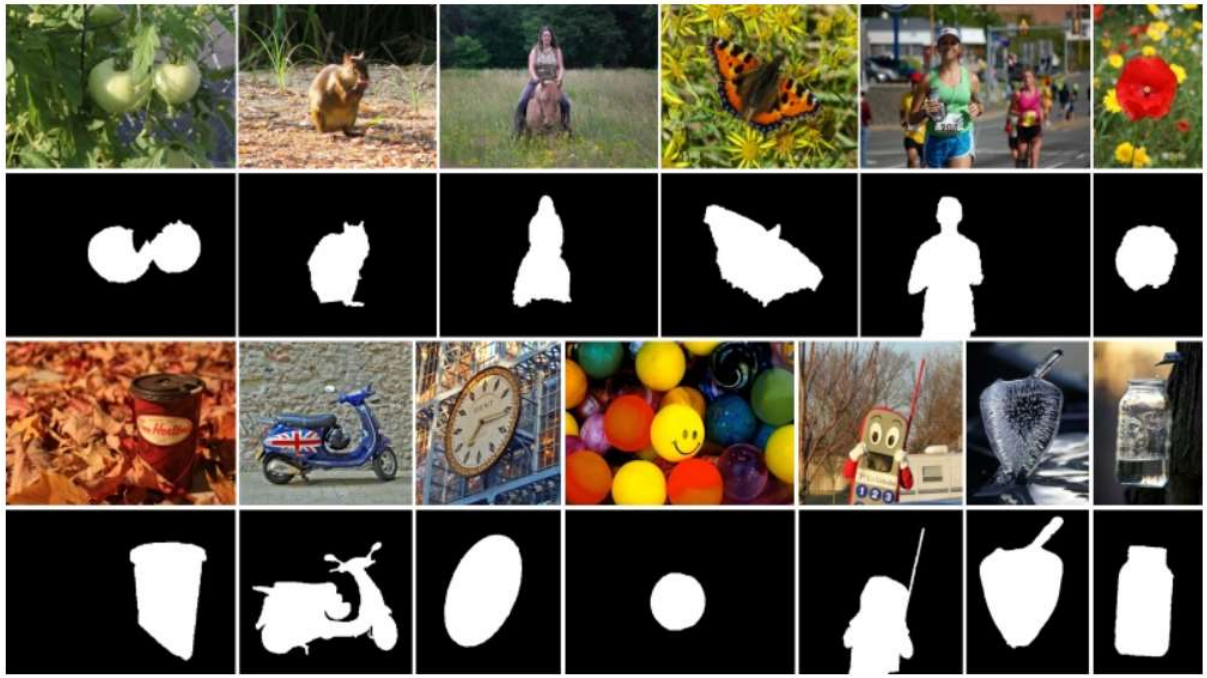
Next Steps

- Continue with saliency detection
  - DeepL
  - CNN
  - Find dataset

Next Steps



[1]


[2]

Sources

[1] https://www.kaggle.com/jessicali9530/mso-dataset#COCO_COCO_train2014_000000017429.jpg

[2] http://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/dataset.html

[3] https://www.pyimagesearch.com/2018/07/16/opencv-saliency-detection/

Make It Aesthetic

Anna-Maria Hohmann
Konrad Kring
Meret Lindanis

5th presentation
17.12.2019