

Research work - Report

- Rakstu darbi Ģeneratīvo sacīkstes tīklu nodaļā. ✓

<https://www.overleaf.com/project/61cdb1a01f9347fe4d1e3433>

- Samples that are filtered from fer dataset using segmentation masks:

Check if all three masks are present and how much of their pixels exist. For average good mask examples it should be > 200 mask pixels.

```
def check_masks_exist(masks: np.ndarray):  
    masks_exist = False  
    eye_mask = masks[0]  
    eyebrow_mask = masks[1]  
    lips_mask = masks[3]  
  
    eye_mask_px_count = len(np.argwhere(eye_mask > 0.5))  
    eyebrow_mask_px_count = len(np.argwhere(eyebrow_mask > 0.5))  
    lips_mask_px_count = len(np.argwhere(lips_mask > 0.5))  
  
    if eye_mask_px_count > 200 and eyebrow_mask_px_count > 200 and  
    lips_mask_px_count > 200:  
        masks_exist = True  
  
    return masks_exist
```

- Example samples that were filtered out using above method (mouth not visible, part of eyes/closed eyes, random images, too small features in sample etc):



- Use mask classifier to filter target emotion from source, and also other emotions.
- Example filtrations from source-neutral:



Feature-to-feature:

- All features cropped and scaled:

Example original image and cropped/scaled ROI:



- After forward pass through model put generated ROI back in image, and apply filter mask:

```
def overlay_image_alpha(img, img_overlay, x, y, alpha_mask):
    """Overlay `img_overlay` onto `img` at (x, y) and blend using
    `alpha_mask`.
    `alpha_mask` must have same HxW as `img_overlay` and values in range [0,
    1].
    """
    # Image ranges
    y1, y2 = max(0, y), min(img.shape[0], y + img_overlay.shape[0])
    x1, x2 = max(0, x), min(img.shape[1], x + img_overlay.shape[1])

    # Overlay ranges
    y1o, y2o = max(0, -y), min(img_overlay.shape[0], img.shape[0] - y)
    x1o, x2o = max(0, -x), min(img_overlay.shape[1], img.shape[1] - x)

    # Exit if nothing to do
    if y1 >= y2 or x1 >= x2 or y1o >= y2o or x1o >= x2o:
        return

    # Blend overlay within the determined ranges
```

```
img_crop = img[y1:y2, x1:x2]
img_overlay_crop = img_overlay[y1o:y2o, x1o:x2o]
alpha = alpha_mask[y1o:y2o, x1o:x2o]
alpha_inv = 1.0 - alpha

img_crop[:] = alpha * img_overlay_crop + alpha_inv * img_crop
```

Example for inserting ROI back in to original image (transfer is bad for this run, still training).



ROI image brightness changes, probably because of white/black face sample dominance in that particular batch. Need to train longer for generator to learn difference?

- **Separate features. Mouth region and eyes.**

Already modified cyclegan model and preprocessed data. Will run this next. TODO

Next tasks:

Filter Celeb dataset for neutral and happy emotion. There is not really different emotions there, but can filter other from Flickr faces dataset.

Train face-to-face with RGB images filtered from previous step.

...