Project Report - Johnny Borkhoche

The goal is to produce a state-of-the-art on different approaches that already exist for face detection. I will also try to develop some type of metric to compare said approaches based on their requirements in hopes of classifying them into grades.

For detecting facial landmarks, it would be interesting to find a single method that would allow the detection of all facial features, i.e. ears, hair, eyes, cheeks, nose etc. since so far singular techniques have been only detecting part of the face.
The overall challenges that are faced when detecting faces are the following: unusual expressions, illuminations, skin types, distance, orientation, complex backgrounds, many faces in one image, face occlusion, low resolution.

I also wish to try and understand what are the different parts of the pipeline we are modeling, so that I know what are my options for what to work on.

Known technologies :

- Dlib:
    - Open-Source, well documented toolkit for ML algorithms using C++. I don't know what algorithm is currently being used but they have a pretty comprehensive list
- Google Face Mesh:
    - their paper
    - Seems like the most comprehensive facial detection tool (captures pretty much all of the face). But integration with other parts of the pipeline might prove very difficult

What I've found :

Facial Landmark part of the pipeline

- Article A review of image-based automatic facial landmark identification techniques
    - This article pretty much reviews methods for face recognition used over the years
    - Foundational methods for face detection that can be used off-the-shelf from OpenCV and Dlib (not state of the art anymore):
        - Viola-Jones : -> not used very often, more of a reference method
        - Histogram Of Gradients (HOG) : Image is converted into a series of histograms based on the orientation and magnitude of pixel gradients within the image. A SVM classifier is used to identify the face based on the histogram values. *Good thing* about it is that it has low power consumption, no need for GPU
        - Results of comparison between both models: The Dlib, HOG-based face detector outperformed all of the OpenCV variants with greater accuracy and fewer false positives. Both the Dlib and OpenCV face detctors performed better when applied to images from controlled environments.
- Article A comparative study of face landmarking techniques
    - Landmarks are categorized into 2 here: Primary (inner eyebrows, nose tip, eye centers, mouth corners etc) and Secondary (outer eyebrows, chin tip, temple etc)

    - Some pre-processing is imperative before engaging in landmark detection, such as illumination, modest geometric corrections, segmentation of the face, use of color information..
    - Two basic categories of facial landmark detection:
        - Model-based methods, considers the face image and the ensemble of facial landmarks as a whole shape. The mode learns 'face shapes' from labeled training images, then at the testing stage they try to fit the proper shape by an unknown face
            - Explicit Methods (Active Shape Model, Active Appearance Model)
            - Implicit Methods (algorithms using neural network applied to the whole face)
        - Texture-based methods, aims to find each facial landmark (or local groups of landmarks) independently, without the guidance of a model.
- Some potentially useful 2D datasets (last part of parag. 5 -> CPU runtimes not satisfying for deep-learning methods):
    - Multi-PIE: 337 subjects in 15 views
    - XM2VTS: 295 subjects, 4 recordings of each
    - 300-W: 300 faces in the wild challenge dataset, 3837 images some indoor some outdoor
    - Menpo: 6679 semi-front view face images, 5335 profile view images, all annotated
-

Other parts of the pipeline: how does the robot draw?: image processing + transformations?

- Paper Making Robots Draw A Vivid Portrait In Two Minutes : AiSketcher
    - interesting website associated with it and github but *NO CODE*
    - their algorithm looks like this:
        - First part: pre-processing -> face detection, alignment, parsing + background removal
        - Second part: global sketch synthesis via NST + local sketch synthesis via GAN
        - Third part: post-processing -> image binarization + fusing global & local sketches + eyeball renewal + chin renewal
    - They end up with a synthesized sketch, more cartoonish. Our current results are actually better looking than these
- There doesn't seem to be many projects online where robots are progammed to draw human portraits.
    - Only one project from `Patrick Tresset` keeps recurring but with no code/paper. Only videos. Our results look better

Idea(s) :

- A good way to process might be to computationally generate the human being's *average face* (openCV guide here) to use as a base. Then, when we wish to caricaturize a person's face, we find the area of the face which has the biggest difference (pixel-wise) to the average face and choose to this area to focus on for caricature. The goal is to try and imitate how real caricaturists would reason when drawing portraits. I am proposing this idea because it felt like some of the existing architectures (such as WarpGan) are too local when they do their transformations. (Should I try to implement a basic version of this?)
    - Challenges faced: a flagrant issue is that the generated average face is heavily dependent on the input data, so if we mainly supply white male faces then we will very

likely end up with a white male face. To be more inclusive, we must have a balanced dataset with equal parts of different ethnicities, or calculate different average faces for different backgrounds (depends on how we want to implement)

- *NOT VERY USEFUL* Another way to look at it is if we use the picture of the client as an input to a GAN (idk if that's possible) such as StyleGAN to generate a face similar to the initial one and then draw it. So this is like a pre-processing step to change how the face looks like, but not really in line with the caricaturist spirit.

Not exactly related but potentially useful

- OpenPose
  - First real-time multi-person system to jointly detect human body, hand, facial, and foot keypoints (in total 135 keypoints) on single images.
- Paper Learning Perspective Undistortion of Portraits
  - Long story short: the closer the camera to your face when taking a picture of it, the higher the rate of perspective distortion -> creates a face with unusual proportions.
  - They try to remedy this issue by introducing a pipeline that uses 2 networks to rectify distortion and inpaint the missing facial features.
- If we require datasets, there are quite a few listed at the bottom here

After thoughts

Seems like most of my research ended up with finding Machine Learning models that require training on annotated datasets... this might not be what we want to go for (I remember being told that we might not need to do that, but I am unsure regarding the specifics.)
I also tried finding information on how to transform human portraits (image processing I presume) into something that robots can draw (is it done through curves?). I couldn't find much online (surprisingly, or I just didn't know how to google appropriately) so I need to see how it's currently being done in our project.

Overall I feel like I would like to either try to implement a small version of my first idea if it seems to have any potential, or I would like to understand (And maybe work on) the image processing currently being done to transform the portrait image into what the robot is actually drawing (what's the logic behind it, is it simply some mathematical transformations? does it use curves to highlight parts of the face? etc). As I feel like I don't have a holistic view of what is going on in its entirety (regarding the different parts of the pipeline), I would be open to amendments/suggestions regarding what you think I should work on considering my level.

Easter-Egg

I ended up finding Mr Calinon's paper from all the way back in 2005!