CPS803 Group 3: Project Proposal

Nicolas Trivanovic
Computer Science
Ryerson University
Toronto, Canada
nikolas.trivanovic@ryerson.ca

Craig Pinto
Computer Science
Ryerson University
Toronto, Canada
craig.pinto@ryerson.ca

Ahmet Cengiz
Computer Science
Ryerson University
Toronto, Canada
acengiz@ryerson.ca

Mohammed Qasim Khan
Computer Science
Ryerson University
Toronto, Canada
mohammedqasim.khan@ryerson.ca

I. Introduction and Motivation

The prevalence of public surveillance raises concerns for the privacy of citizens and instances of potential noncompliance with anonymity laws for corporations. Cases where monitoring people or locations without logging facial data that can identify a person i.e. running a mass public survey, crowd analytics, or alternatively law enforcement situations where recording personal data would be biased.

Our goal is to develop deep-learning algorithms that we learn through the CPS 803 course that can detect a face and subsequently apply a dynamic face-blurring filter. Our algorithms should be able to do this for both still and live images.

II. Method

We aim to use a dataset that includes sets of face and non-face images for training and validating our face detection algorithm with cases of false positives and false negatives.

We aim to use 3 different machine learning algorithms for detecting and prioritising facial data which are: ANN, K-Nearest Neighbour, and SVM.

Our face blurring filter will be attempted to be dynamically applied, so facial data cannot be reconstructed. We will use a generated kernel convolution that will not warp the dimensions but rather the pixels inside the boundaries set by our face detection.

The eventual goal is to set up an application that will open your front-camera and blur your face live.

A. Data Constraints and Preprocessing

We are going to be detecting mainly clear-face (no sunglasses, face masks, etc) images in our training and validation sets. We hope that we can properly detect images with facial accessories or obstructions at a suitable rate but if it is a problem we will develop our algorithms for clear-face images only.

We will add noisy images in our training and validation sets to reflect the poor quality of most surveillance cameras. We will try to have our face detection algorithms work well for these images as well.

Preprocessing will include using denoising filters (Gaussian), illumination normalization, and adaptive thresholding to increase the "clarity" of images. We may also have to include homography to account for different poses and angles of faces in images and to flatten the image.

B. ANN Algorithm (1)

We will base our ANN on HOG detection, which will detect gradients across our image. It will run two convolutions across the horizontal and vertical axes of our images based on pixel brightness to capture contour and edge information.

We will then subdivide the images into cell blocks and compute the histogram of gradient direction of each block, which we will then normalize in comparison to surrounding blocks.

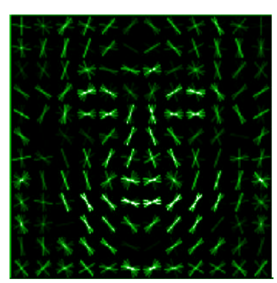

Fig. 1. An image after being ran through HOG

This will turn the image into a representation of gradient vectors (Fig. 1) that we can assign weights to depending on their directions. This can isolate features like the eyes and nose and “extract” them to train our ANN. Our ANN will then be run several times to improve its prediction.

C. K-Nearest Neighbor Algorithm (2)

We will first perform the extra preprocessing of “face window” setting and cropping, as well as feature extraction using Principal Component Analysis to offset the false positives/negatives that would otherwise arise from running KNN directly.

We will then run our KNN classifier on a well distributed set of face and non-face images to allow the algorithm to separate them into clusters. We will tweak our algorithm to have only two clusters (face vs non face) rather than the multiple clusters that could arise.

D. SVM Algorithm (3)

We will use canny edge detection and change-in-colour gradient (skin colour) to define the boundaries of faces in our images.

We will perform feature extraction by calculating the HSV difference around these boundaries to detect the eye-whites and check against pupils and confirm the presence of eyes in the captured area. We will also detect lips using a similar method. Lastly, we will detect the nose using the position of the eyes and lips and subdivide this area checking for vertical edges.

If these features are found, then we will confirm the image boundaries as containing a face and then classify it as such. If not, then our SVM will classify it as not a face.

E. Blurring Algorithm

Our blurring algorithm will be a dynamically generated kernel that we will convolve on the faces captured by our 3 algorithms.

We will maintain the boundaries of the face area and only manipulate pixels within them to produce our blur effect.

III. Intended Experiments and Observations

We will divide our dataset of face and non-face images into a training set containing 80% of the images and a validation set containing the other 20%

We will check the precision of our algorithms in order to improve our algorithms and populate our report.

We intend to record any interesting observations or trends in data we see into our final report. We will also note the processing and run times of our algorithms.

Finally, we intend to create a simple application that will open and use a webcam to demonstrate live face-blurring.

IV. Project Schedule

Since the project can be broken down into two parts of face detection and the blurring algorithm, we will firstly complete the face detection part. We are using 3 separate algorithms for face detection, so we will try to work on them in parallel.

WEEK [4-7]: Developing our preprocessing algorithms, working on our ANN
WEEK [8-10]: Working on our KNN and SVM
WEEK [11-13]: Testing, observations, and completing our final report. Preparing demo application.

V. References

[1] Shah, Parin M., "Face Detection from Images Using Support Vector Machine" (2012). Master's Projects. 321. DOI: https://doi.org/10.31979/etd.wg5s-gyqn https://scholarworks.sjsu.edu/etd_projects/321

[2] Nafchi, H. Z., & Ayatollahi, S. M. (2012, January 1). *A Set of Criteria for Face Detection Preprocessing*. ScienceDirect. https://www.sciencedirect.com/science/article/pii/S1877050912007326

[3] Tyagi, M. (2020, May 5). *Face Recognition Using Knn & OpenCV - Analytics Vidhya*. Medium. https://medium.com/analytics-vidhya/face-recognition-using-knn-open-cv-9376e7517c9f