CDA571: Facial Detection and Beyond

Mark Gordon
Department of Computer Science and Engineering
University at Buffalo
markgord@buffalo.edu

Georgius Aland Abibenar Feltama
Department of Computer Science and Engineering
University at Buffalo
georgius@buffalo.edu

Christopher Wade
Department of Computer Science and Engineering
University at Buffalo
cwade@buffalo.edu

July 24, 2020

1 Project Details

Team Name:	Team 2
Project Name:	Facial Detection and Beyond
Team Members:	Mark Gordon; Georgius Aland Abibenar Feltama; Christopher Wade
YouTube Presentation:	https://www.youtube.com/watch?v=Ln1bzUpu9Mk

2 Problem Statement

Our team is performing facial detection. We set out seeking to explore how facial detection works and potentially find areas to improve upon. The reasons for our interest in this technology are manifold. In our society, law enforcement access to this software has been recently restricted. There are many issues creating biased outcomes within facial detection, which has led to discrimination. This has been proven to exist within current iterations of facial detection software, which has created a negative impact among communities of color. The depth of the problem inherent in this technology is difficult to gauge, as the exceedingly complex neural networks which drive facial detection and recognition operate in a metaphorical black box. There are a couple of primary issues that we ran into, such as computational limitations and time

constraints. Traditionally, facial recognition and detection software has been run on large servers with multiple dedicated GPU's and ram to spare. Working from our homes with limited ram and single GPU's forced us to reconsider the methodology utilized in tackling the problem.

3 Dataset

The data set we chose to use, called *WIDER Face*, is part of an Amazon object detection and recognition database and can be downloaded from Kaggle. The data set comprises 34,000 images with 394,000 known faces. The images span 61 different categories ranging from sports, parades, riots, and police raids. Although they are separated by category, the sorting is imperfect and it is not uncommon to find misclassified images.

In the training set, there are 12,880 images with over 150,000 bounding boxes, averaging 12 faces per image. Many of the faces are difficult to detect: about 85% of the faces in the training set are blurry, 19% of are partially occluded (slightly blocked), and 24% are almost fully blocked. Additionally, 1,845 of the faces have an unusual expression with and 6,042 faces are arranged in an atypical pose. Some of the faces (1.6%) in the training set have been marked as "invalid" due to the impossibility of detection and have been removed in preprocessing. Moreover, faces smaller than 20 square pixels have been excluded from the training data due to the difficulty of accurately identifying a subject with such little data.

The test set contains approximately the same amount of images, but bounding box coordinates and additional information are not provided due to the nature of the challenge. As such, we made use of parts the training and validation set for both training and evaluation. The test set can still be used to gauge generalization performance at a base level, as it is fairly easy to draw the bounding boxes on the images and evaluate the successes and shortcomings.

4 Method

To pre-process the data, marked instances of invalid bounding boxes were stripped from the data set. Then, instances of bounding boxes which were deemed below the threshold of determinability were stripped as well. Threshold of determinability was defined to be a total area of bounding box below one. The images these bounding boxes were removed from are left in the data set.

The data was read into an intuitive database created by Mark Gordon, where the data was able to be stored with limited burden on local memory and accessed easily. He then created a cascade classifier object to run HAAR cascade classifier models so we could begin testing on our data set without reinventing the wheel. At this point in the project, we quickly realized that the time constraint of running these pre-built models would not allow us to take our project to the initially intended goal. If a single run through of a fraction of

the data could take up to thirty minutes on a pre-trained model, then training a model even on a pre-built network would inevitably fall outside the time limit for the project. Training a model typically involves randomly running through the training data hundreds, if not thousands, of times to give the network ample opportunity to establish strong predictors which are robust.

After testing multiple precomposed models which had been trained on other data sets, we instituted a transfer learning approach. The approach centered on multi-task cascaded convolutional networks, or MTCNN. MTCNN is an agglomeration of smaller neural networks; specifically P-net, R-net, and O-net. P-net determines bounding boxes for candidate faces and calibrates those candidates based off regression vectors before merging extensively overlapped candidates. R-net then rejects false candidates before repeating the the secondary and tertiary tasks of P-net. Finally, O-net exports finalized outputs of five land marked facial positions [1].

In order to implement this, it was first necessary to take a subset of the training data. To ensure robustness while attempting to minimize run-time, we opted to use 5,000 positive (contains a face) and negative (does not contain a face) samples. Generating this data is fairly straightforward given the provided bounding boxes for training data. Bounding boxes that had an area greater than 25 pixels are selected at random, and the face is cropped from the image to create a new image. Smaller faces could be used for training, but they are less likely to have a positive impact on the model. To obtain negative samples, an image is selected at random and a coordinate within the image is chosen. If the coordinate does not correspond to a bounding box, a sample of 48x48 pixels is taken and used as a negative. Each of the positive and negatives samples need to be re-scaled to 12x12, 24x24, and 48x48 for the P-net, R-net, and O-net respectively. After they have been re-scaled, the images are fed to the appropriate network with the corresponding class label. After the process is done (which takes approximately a day per network), the resultant weights are saved back into the original model.

Testing for accuracy was accomplished with the usage of an intersection over union metric. We could not find any sort of package which had this made for us, though one likely exists, so Mark just wrote one based on documentation. The intersection over union metric, or Jaccard index, is popular for image segmentation and object detection because it quickly compares true bounding box location to predicted bounding box location. As the name implies, it takes the intersection area of the two bounding boxes and divides their combined area. This results in a number between 0 and 1, with 1 being a perfect match. Typically, a score above .5 is considered reasonably sufficient, but for very small boxes, minor translation or scale errors can prove very detrimental. Additionally, the models were not trained exclusively on the relevant data, leading to slight inaccuracies in what constitutes the full face (excess hair, for example). For this reason, some lenience was granted to what was considered an acceptable score for these instances: anything under .2 was considered bad, while scores under .3 were considered unlikely but possible.

In addition to evaluating matching bounding boxes, it is also possible

that the predicted and real numbers differ. In the case that there are fewer guessed bounding boxes than real boxes, these are considered misses or "false negatives". Should there be more guessed bounding boxes than real boxes, these are explicit "false positive". The distance between each guessed box to each real box is calculated, stored, and sorted. The closest pairs are chosen in order and removed from the available set of pairable boxes. Note that if boxes that are paired are far away from each other due to a situation where the same number is guessed but a face was missed *and* something was falsely identified, they will be a non-explicit "false positive" in the form of a bad match (low or zero IOU score).

5 Results

Utilizing the pre-compiled Haar Cascade classifiers, without training on our specific data set, we were able to achieve a 16.62% accuracy rate overall with high confidence. This works out to 73.7% of detected bounding boxes being accurate. Reduction to gray-scale, while improving run times significantly, reduced accuracy significantly as well. Upon institution of the MTCNN transfer learning approach, accuracy was increased to 29.1% overall with 94.7% of detected bounding boxes being with high confidence.

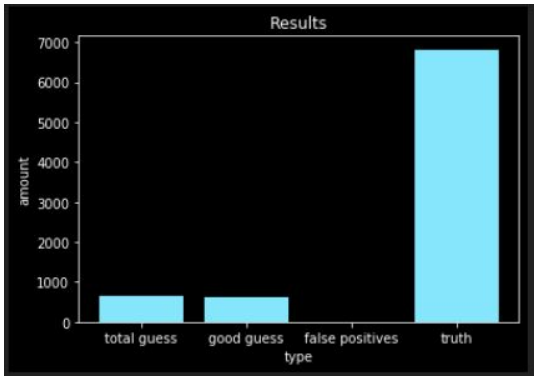

Figure 1: Precomposed CNN (Before)

We were also able to implement the transfer learning model in real time using a video capture device through *OpenCV* in order to draw bounding boxes on our own faces at 30FPS (with some notable lag that goes away, but makes the video less smooth, at 15FPS). It was able to detect our faces with partial covering, but experienced difficulty if a large portion of our faces were obstructed, especially both eyes. This is because the model learns to recognize key points within the face, such as eyes, nose, and lip edges to determine a positive identification. It uses this knowledge as a basis for confirming matches, which illustrates the inherent difficulty in occlusion, blurriness, and irregular angles or poses. In addition to our own faces, it was also able to identify faces of people

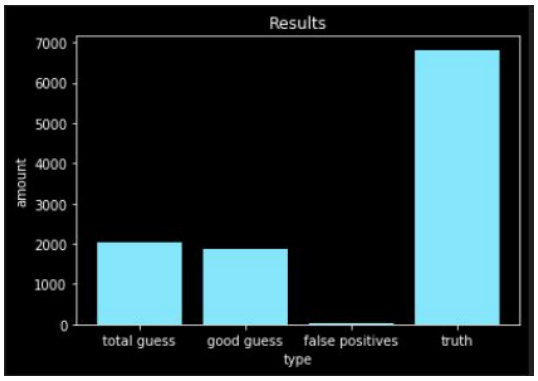

Figure 2: Transfer Learning (After)

in pictures displayed to the webcam - both physical and virtual - with relative ease. Overall, it is interesting to observe the results of transfer learning and it is fun to think about the prospect of training a model this complex from scratch with the required resources.

6 Feedback

During the course of our work, we were offered a great deal of feedback that helped us progress effectively in a timely fashion. We were strongly encouraged to take advantage of as many methods of dimensionality reduction as possible. Some suggestions included the use of PCA or Fischer vectors. As it turns out, the package that we utilized, openCV, already incorporates both PCA and a derivation of Fischer vectors, known as Fisherfaces, internally. Instead, we sought other optimizations through the use of GPU-enabled processing and multi-threading through *OpenCV* and the use of grayscale conversion. Figuring out how to make this run fast without sacrificing the quality of the model or evaluation has been a priority. In the interest of time, we began sampling the test data both by 5% and 10% to sufficiently generate results for analysis. With more time (and computing resources), this would be a much more approachable endeavor.

7 Conclusion

With an appropriate amount of time, computing power, and dedication, accurate and effective high speed facial detection is achievable. We were unable to delve into the greater issues which led us to tackle this project, as time limitations simply cannot be circumvented. It is not outside the realm of possibility or even probability to believe however, that facial recognition can be adjusted to handle inherent biases in the future. Given a data set which has been carefully

cultivated to include a diverse variety of persons from all races and genders, it is probable that a model can be trained which holds as good indicators markers which have not been reinforced in previous models. In the event that this is too much for current systems to handle, it has been shown that a transfer learning approach can be employed to workaround the limitations of using a single model.

References

[1] Kaipeng Zhang et al. *Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks*. Accessed: 7-13-2020. URL: https://kpzhang93.github.io/MTCNN_face_detection_alignment/paper/spl.pdf.