Automated Feature Labelling: Head Detection in Film Clips

Anirudh Satish
Harvey Mudd College
asatish@hmc.edu

Abstract

This project seeks to perform head detection and localization of all heads in the frames of short video clips. This not only includes faces, and heads that are clearly visible from the front view, but also heads and people from different angles and profiles. This is achieved through a Deep Learning approach, implementing a pre-trained SSD based model on the Gaze Data set (2) (5). The code can be accessed at my public Github repository here

1 Motivation

Object Detection has been a field of great a interest to the Computer Vision community for decades, with the earliest Neural Network approaches beginning all the way back to the 90s (7). Despite this early start with deep learning approaches, advances in this field were relatively slow until CNN and R-CNN approaches were developed along with support vector machine (SVM) architecture (7). While Face detection has come along way, with many of us unlocking our devices just by looking at a blank screen, the same cannot be said about the head detection subspace of object detection. The prevalence of state-of-the-art face detection applications can be seen all throughout the technology space, and even the very simplest of models can perform to staggering levels (6). For head detection and localization, state-of-the-art object detectors achieve on an upper bound only around a 65% Average Precision for people by the Pascal VOC (3) benchmark (8) (7).

The challenge that this sub field of object detection poses is one of the driving forces of motivation behind this project, which is to localize every head visible in short movie clips. There are a lot of reasons for these challenges. One such reason is that detecting heads is hard as the identifying features of a head are not so clear cut when compared to something like a face, that has very precise features such as eyes, nose, ears and more. Head detection is particularly difficult when the head is facing away from the camera, and only the back profile can be seen. Another difficulty that faces projects of this kind is that the human pose can take a lot of different forms in videos, and occlusions and other objects often block key features in the detection process, which have added adverse effects due to the minimal number of identifying features surrounding head detection (8). Understanding these obstacles, and trying to get a working system that performs well by getting around these difficulties is a huge motivator for me. While this project is treading into unknown territory for me personally, the research that has been done in this field, particularly in the deep learning space was very useful in being able to understand such systems, and chose the right approach for the task. The Related Works section talks more about these advancements.

This project's overarching goal is to create a method to return the bounding boxes for frames that are part of the Gaze Data (2) provided by Professor Katherine Breeden, Harvey Mudd College. One of the applications of the results of this project is as a stepping stone toward developing a system that detects cuts and shots in movie sequences. The goals of Prof. Breeden and this scope align well with my interests and the shared motivation resulted in this project.

2 Related Works

State-of-the art systems for head detection as expected make use of Deep Learning methods. Tuan-Hung Vu *et. al* in their paper "Context-aware CNNs for person head detection" (8) use three different Neural Networks, which all work together to provide the bounding boxes around the heads in frames. The big picture of this architecture is to use contextual information in the frame, as an additional set of features, along with the spatial location of the detected objects to reason about Human Heads and detect the same.

A less computationally intense framework follows an implementation of the Single Shot Multi-Box Detector, an object detector, proposed by Wei liu et. al in their paper (4). This architecture uses a single Neural Network, and achieves speeds of 59 FPS with mAP 74.3% on VOC2007 test, vs. Faster R-CNN which achieves only 7 FPS (4). With this huge speed boost, naturally the accuracy of this model is not at state-of-the-art level, but it is still high enough achieving 80% accuracy on standard tests (4). Manuel J. Mar´ın-Jimenez et. al in their paper titled ”LAEO-Net: revisiting people Looking At Each Other in videos” (5) use this SSD architecture to build a head detector to identify people looking at each other. The authors of this paper used this SSD head detector model to supplement their goal of detecting people looking at each other in film videos. The alignment of the aforementioned paper’s implementation with my project allowed me to re-purpose their model for my task of simple head detection.

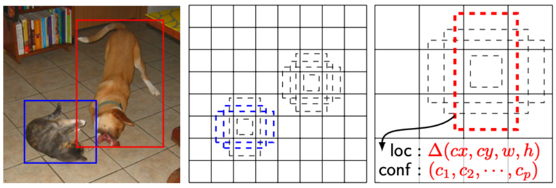
3 Methods

3.1 Data Set

The data set that I have for this project is 15 short film clips, of around 1-4 minutes long. This data set was annotated by rigorously recording the gaze behaviour for 21 participants from ages 22-73. These videos were split into frames and each video roughly had around 4000 frames. Note that this is the data that the detection needs to be done on, not trained on. For the task of head detection, most of this annotated data is irrelevant to evaluate against, since there are no ground truth labels for the presence of heads (not faces) in each frame. However, since the presence of a face indicates that there is also a head, I use the labels of the presence of faces as a pseudo quantitative evaluation metric.

3.2 SSD Model

The model I used for this task was a pre-trained Single Shot Multi-box Detector (SSD) that was trained on the ’Hollywood Heads’ data set (1). This data set contains 369,846 human heads annotated in 224,740 video frames from 21 Hollywood movies, and is a widespread data set used by the Computer Vision community in training their Models. This pre-trained model was trained with a Learning rate of 10^{-4} for the first 50 epochs, and a factor of 0.1 for the rest of the training procedure (5). This training procedure also incorporated batch normalization and data augmentation for robustness and improvements in speed (5). The model expects input frames of dimensions 512 x 512, so one option is to reformat the input to meet these dimensions and translate back the results after predictions, or to retrain/reformat the model to expect inputs of required size. However, due to the variability in input image sizes, the former is the better and more reliable approach.


loc : $\Delta(cx, cy, w, h)$
conf : $(c_1, c_2, \cdots, c_p)$

(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Figure 1: Visualization of the training procedure for the SSD based detector.

The SSD model is simple in that it only requires an input image and ground truth boxes during the training process (5). The way the model works is as follows: During training, the model evaluates a set of pre defined bounding boxes of different aspect ratios at every possible object location which is determined by feature maps (5). These predicted boxes are matched against the truth labelled boxes and the one with the least loss is chosen (5). The model’s loss function is a weighted sum between a localization loss function and a confidence loss function (Softmax) (5). During Prediction, the same pattern follows, except there is no ground truth to measure against, and this is where the model weights come into the picture to get the best prediction.

3.3 Finding Bounding Boxes

To compute the bounding boxes/coordinates of the same, each individual frame is read in, and converted to a reformatted array representation, which essentially means it was resized to match the required 512 x 512 dimensions. Following this, the array representation of the image is fed into the model, and a list of coordinates for bounding boxes is returned (0 if none). These coordinates were used to compile the results in two formats.

The first, a simple CSV file for each clip, where each row corresponds to a frame, and

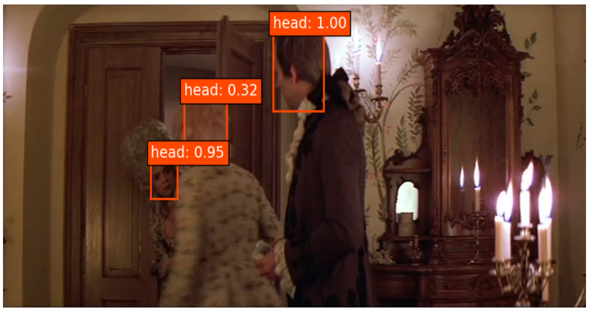


Figure 2: Clip : Amadeus

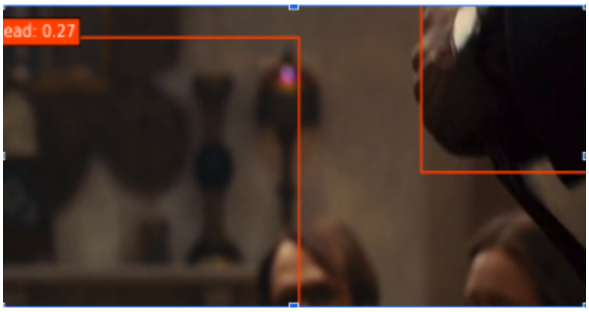


Figure 3: Clip : Argo

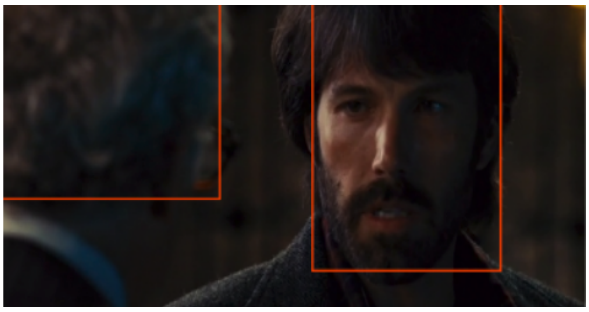


Figure 4: Clip : Argo

columns 1 to n (indexed at 0) each represent a bounding box in the frame in the format $([xmin, ymin, xmax, ymax])$. This representation is much more space efficient, and can be used for quantitative evaluation, or any other purpose as one pleases as long as the correct row from the CSV file is matched to the correct frame.

The second representation is simply creating a new .png image for each frame that was predicted, with the bounding boxes visualized. This is useful if you wanted to stitch together the frames to a video, or for any qualitative evaluation.

There are two scripts written in Python that can be run to get whichever output is desired, with the added functionality of being able to run on all the video clips at once using a wrapper, or any single video frame, using command line arguments. Creating the CSV files for all the clips took around 2.5 hours, running on 2 GPU's. Creating the bounding boxes and saving them to a png image takes a bit longer, around the order of 4 hours for the whole computation.

4 Results and Evaluation

Snippets of the results from using this model on the Gaze Data set are illustrated below. To access the complete set of CSV results for all the clips, refer to my Gitub repository, under the folder titled OutputCsv in the Master branch. To evaluate the performance of the model, I do both a Qualitative and Quantitative assessment.

4.1 Qualitative Evaluation

For this evaluation, I merely looked at around 50 random predicted bounding box frames for 4 movies, Amadeus, Argo, Kings, Gladiator and observed how accurate the predictions were. The factors that I looked for were if clear faces were detected as heads, if occluded heads were detected, if multiple heads were detected in a frame if necessary, and finally the quality of the bounding boxes that were produced.

In summary, the results were good. Clear cut faces were always detected as heads with a confidence of 1, or very close to 1. Most side profile heads were detected with confidences upwards of 70%. The predictions and bounding boxes were inaccurate, sometimes rarely non-existent when the features for heads were minimal (back profile as an example).

4.2 Quantitative Evaluation

Due to the lack of ground truth labels that indicate the presence of heads or not, I used the labels for the presence of faces as a pseudo evaluation metric. I expected the accuracy for this metric to be very high due to my qualitative observations and just the fact that face detection is easier and that a face

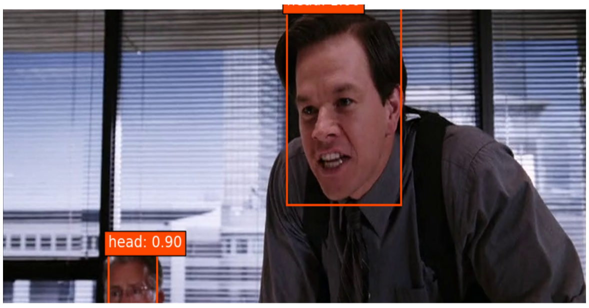
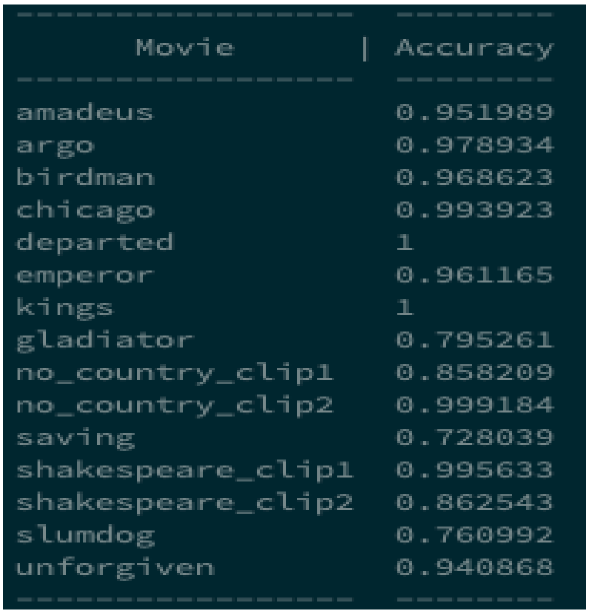


Figure 5: Clip : Departed



```
 -----------------    --------
       Movie        |  Accuracy
 -----------------    --------
 amadeus               0.951989
 argo                  0.978934
 birdman               0.968623
 chicago               0.993923
 departed              1
 emperor               0.961165
 kings                 1
 gladiator             0.795261
 no_country_clip1      0.858209
 no_country_clip2      0.999184
 saving                0.728039
 shakespeare_clip1     0.995633
 shakespeare_clip2     0.862543
 slumdog               0.760992
 unforgiven            0.940868
 -----------------    --------
```

Figure 6: Quantitative Results Summary

always indicates a head but not vice versa. The metric used here was the percentage of frames which contained faces that the model predicted to have heads. In other words, for a given clip, Accuracy = $\frac{\text{\#frames detected heads and have faces}}{\text{\#frames that have faces}}$. The quantitative results for all the test clips is tabulated in Figure 6.

5 Discussion

From the qualitative results, it is quite clear that the model is able to clearly predict heads when there is a face accompanying the person (Figure 2,4 5). This makes sense and is explained by the fact that a human face has many identifying features that the model was able to learn during the training process. Notice that the accuracy metric for the films Kings and Departed are a 100%. Upon manually inspecting these clips, I noticed that the people in these clips are all shot from the front, and therefore their faces are visible. This explains the models perfect performance on these clips. Expanding on this, inspecting other clips such as Argo, nocountryclip2, and Birdman, it is very apparent that clips that have faces, whether from a front view, or even a side profile, will have good performance with this model. While this is very good performance, this is expected as mentioned earlier how face detection state-of-the-art systems are very accurate. Nevertheless, it is still impressive how a very simple Deep learning network like SSD is able to achieve such performance.

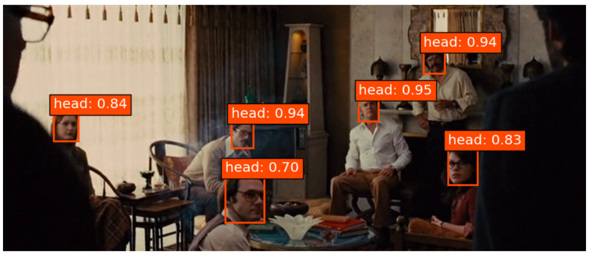


Figure 7: Clip : Argo

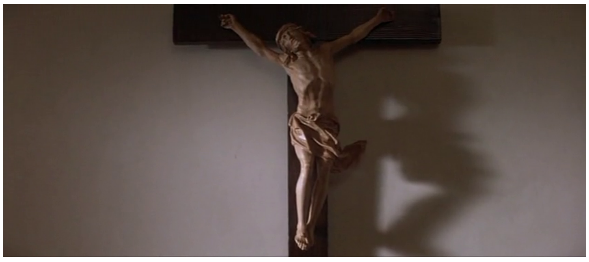


Figure 8: Clip : Amadeus

Another observation is that the model is able to handle multiple heads in a single frame. This is illustrated in Figure 2 and 7. This is particularly impressive given the speed at which the model runs, and the architecture of the same.

Notice that the model incorrectly labels two heads in Figure 3. There are a couple of explanations for this. First, notice that there are two half heads in the frame. Therefore, the model correctly identified the presence of two heads, but was not able to pinpoint the location. This could be due to the presence of minimal identifying features that the model was not able to catch onto. In particular, there are no eyes, and this could be a heavily weighted feature as notice in Figure 5 the model is able to identify a second head in the background, also partially cut off.

Finally, I notice that the model does not perform well on artificial heads. This is illustrated in Figure 7, as the statue's head is not identified as a head. While this can be seen as a pitfall, one could argue that since the model was only trained on human heads, this is outside its expected behaviour, and I intend to side with that argument. However, if someone were to use this model to try and locate artificial heads, this is something to keep in mind. This project, and the model's performance exceeded my initial expectations at the beginning the project. Its performance on detecting even occluded heads was quite good, as can be seen in Figure 2 and 4.

6 Conclusion

I was able to reconstruct a head detector model that I then used to detect all the heads in the Gaze data set, which was the goal of the project to begin. Although this model was not the intended one at the time of the proposal, it turned out to be a good choice regardless, both in terms of performance and the ease of implementation and understanding, not to mention the speed as well.

The pre-trained model was used to make the bounding box predictions, both in .png format, or CSV format, depending on the user's preference. I evaluated my results both by a visual qualitative inspection, and a quantitative metric that made use of the ground truth labels that indicated the presence of faces in each frame.

The main takeaway from the results was its performance on both fully visible faces, and faces that were partially or completely blocked. The performance seems good enough to be able to use on clips for future work. However, on clips that are predominantly back view, the model might struggle. However, to really test this, more data is required as from the few examples of occluded people in the Gaze data, it seemed to perform reasonably.

Future work on this project would be to improve the model's reliability and performance on heads that do not have all features explicitly present. Further, this model currently treats every frame individually, but perhaps coming up with a system that carries forward information and context from previous clips could lead to even better performance. This is an interesting avenue that could be worth researching in. Finally, experimenting with the model's parameters and hyper-parameters is further investigation that can be done for this project.

References

[1] 2011. Vgg. hollywood heads dataset.

[2] Katherine Breeden. Gaze data, stanford university.

[3] Eslami S.M.A. Van Gool Everingham, M., Williams Christopher, K, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *Computer Vis*, 111(98–136).

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. 2015. SSD: single shot multibox detector. *CoRR*, abs/1512.02325.

[5] M. J. Marin-Jimenez, V. Kalogeiton, P. Medina-Suarez, and A. Zisserman. 2019. LAEO-Net: revisiting people Looking At Each Other in videos. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. 2014. Face detection without bells and whistles. In *Computer Vision – ECCV 2014*, pages 720–735, Cham. Springer International Publishing.

[7] C. Monrocq R. Vaillant and Y. LeCun. 2016. International Conference on Artificial Neural Networks (ICANN).

[8] Tuan-Hung Vu, Anton Osokin, and Ivan Laptev. 2015. Context-aware cnns for person head detection.