Guess the Celebrities in TV Shows!!

Nitin Agarwal
Computer Science Department
University of California Irvine
agarwal@uci.edu

Jia Chen
Computer Science Department
University of California Irvine
jiac5@uci.edu

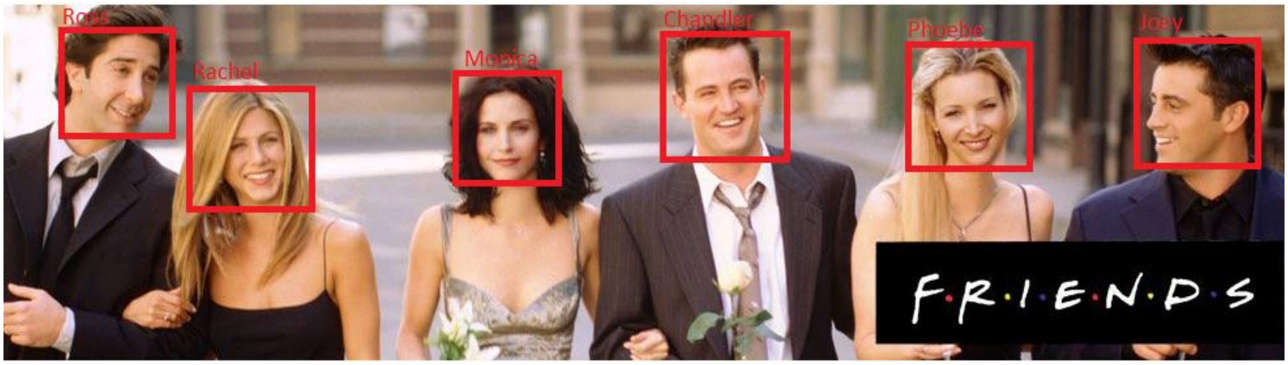


Figure 1: Celebrity faces with character names identified by our system

Abstract

The objective of the project is to develop a system that takes a TV show or movie as input, and automatically detects and identifies all the celebrities shown in it. The system is useful for various applications such as video retrieval, automatic cast listing, etc. The task is more challenging than other face recognition tasks, because (a) compared to static pictures, videos have more motion blur and often poorer image quality, (b) face recognitions in videos require matching a character in various character poses, lighting conditions, facial expressions, makeup and hairstyle etc., which is difficult in most cases.

The project is composed by a training phase and a runtime phase. During the training phase, a classifier based on CNN is trained on 14000 manually labeled faces taken from TV series Friends. In the training data set, each face is assigned a label, which is the name of the character shown in the image. And given a new image, the trained classifier is able to predict the shown celebrity and its probability. We experimented with approx. 13000 faces images and using an off-the shelf CNN with linear SVM achieved an overall accuracy of 72%. We further tested our classifier on the video of Friends movie, which was made in 2014 and achieved an overall accuracy of 49%. Our experiments confirm that good classification results can be achieved just by using an off the shelf CNN with SVM and also that aging significantly reduces the performance of the classifier.

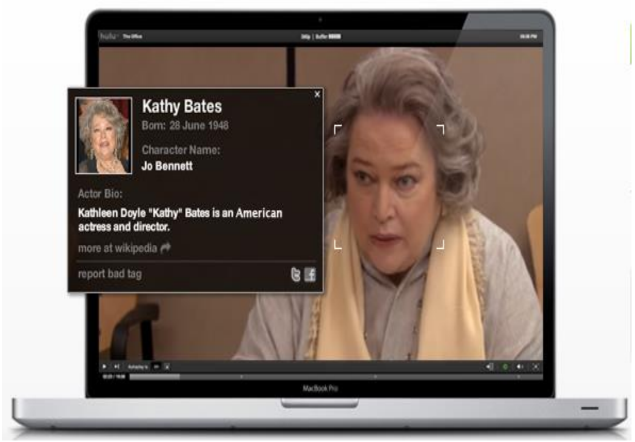

Kathy Bates
Born: 28 June 1948
Character Name:
Jo Bennett
Actor Bio:
Kathleen Doyle "Kathy" Bates is an American actress and director.
more at wikipedia
report bad tag

Figure 2: Hulu Face Match feature

1. Introduction

Associating faces appearing in videos with corresponding names is an important task for many applications, such as video content enhancement, automatic cast listing, video retrieval, video-user interaction, etc. And commercial systems like Hulu Face Mach [1] provide the feature that helps user identify the actors in the TV shows and clips they watch, and provides more information related to the actors if the users hover their mouse over the actor's faces, as shown in Figure 2.

As the commercial systems are based on a large number of video clips and big face data set, they usually require a large amount of human effort to construct and maintain the system. In this project, we explore to design and implement a small scale classifier that is trained on a small number of video clips, with least human efforts. In runtime, the system

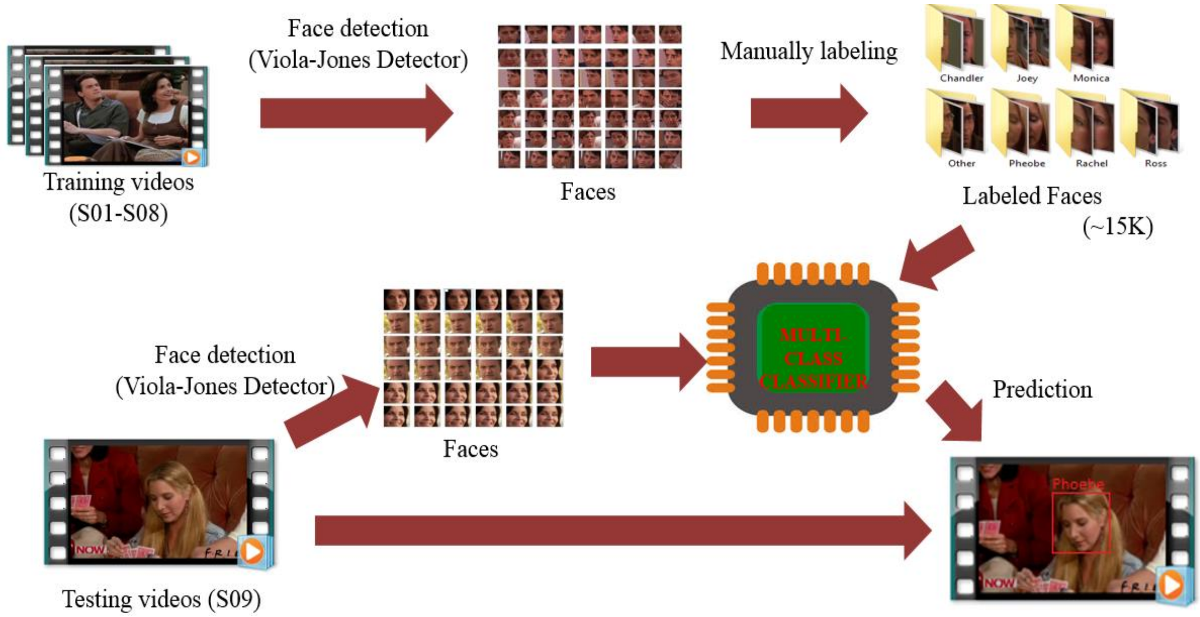


Figure 3: System overview

is able to automatically detect faces from videos, and identify character names from the faces detected.

2. Related work

The majority of methods for identifying names from faces are mainly aiming for static images. One of the earliest attempts was by Kanade [2], who utilized simple image processing method to extract a vector of 16 facial parameters and a simple Euclidean distance measure for matching. Sirovich and Kirby [3] were the first to utilize Principal Components Analysis (PCA) to economically represent face images. Moses et al. [4] proposed using Fisher's Linear Discriminant Analysis (LDA), based on the observation that "the variations between the images of the same face due to illumination and lighting direction are almost always larger than image variations due to a change in face identity". Recently, AI approaches using tools such neural networks and machine learning techniques have been utilized to solve the face recognition problem, which greatly improved recognition accuracy. Samaria and Harter [5] used a one-dimensional HMM to obtain a peak recognition accuracy of 87% on the ORL database. Lawrence, Steve, et al [6] proposed a convolutional neural network based approach to achieve 96.2% recognition rate on the ORL data base.

The task of identifying faces is more difficult when it comes to videos. Because (a) compared to static pictures, videos have more motion blur and often poorer image quality, (b) face recognitions in videos require matching a character in various character poses, lighting conditions, facial expressions, makeup and hairstyle etc., which is difficult in most cases. Satoh Shin’ichi etc. [7] developed a system that associates faces and names in news videos.

The system takes a multimodal video analysis approach: face sequence extraction and similarity evaluation from videos, name extraction from transcripts, and video-caption recognition. Howell and Buxton [8] employed a two-layer RBF network for learning/training and used Difference of Gaussian (DoG) filtering and Gabor wavelet analysis for the feature representation, while the scheme from was utilized for face detection and tracking.

3. System overview

Our system is composed of two stages: a training stage and a runtime stage. As shown in Figure 3, Training stage detects faces from training videos, and trains a model based on the labeled faces. Then based on this model, runtime stage classifies faces shown in testing videos into classes of character names.

Training stage.

(1) *Face detection*: We apply Viola Jones’ Haar cascade algorithm to detect faces from video frame [9]. And to reduce the number of duplicated frames, we extract 1 frame from every consecutive 10 frames.

(2) *Face labeling*: As both the training and testing datasets are based on video clips from TV series ‘*Friends*’, we classify all the faces into 7 classes: ‘Rachel’, ‘Monica’, ‘Ross’, ‘Joey’, ‘Chandler’, ‘Phoebe’, and ‘Other’. And we manually label approximately 14000 training images (2000 per class) of faces as shown in Figure 9.

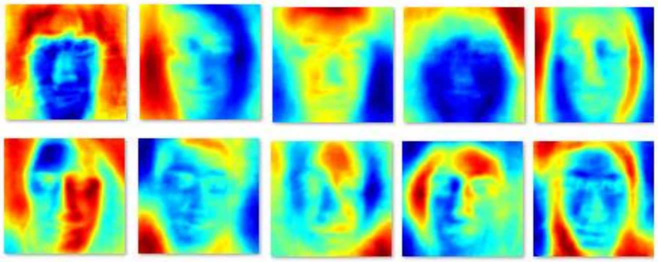


Figure 4: Eigenvectors with top ten Eigen values

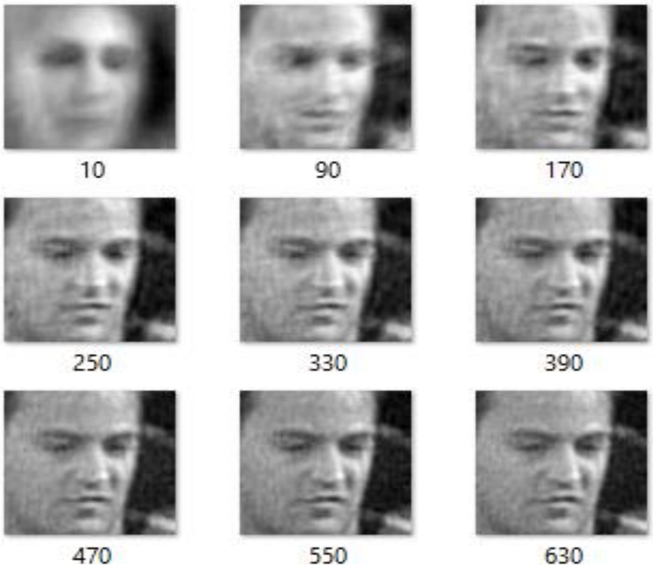


Figure 5: Faces reconstructed with different numbers of Eigenvectors

(3) *Classifier training:* We explore to train two models as character name classifier. One is based on the concepts of Eigenfaces [10], the other is based on convolutional neural networks [11]. Section 4 will describe details of both the models.

Runtime Stage

In runtime stage, we used ~13000 face images from videos of season 9 as shown in Figure 10. The system detects faces using the same face detector we used in training stage, and then the trained classifier classifies the faces into classes of character names. For accuracy estimation we compare the predictions of the classifier with the labeled predictions.

4. Training Models

4.1 Eigen faces with Nearest Neighbour

An image can be considered as a point in high dimensional space, and then the classification problem we are exploring can be converted into a problem of finding nearest neighbor for a given vector. But the naïve idea of considering each pixel will lead to too high dimension, for example, an 80x80 image will be a 6400 dimension vector, which is useless for any practical application. A natural idea for solving these issues is to represent image by a lower dimensional point using concepts like PCA.

Eigenface algorithm [9] converts each image to an eigenvector, and orders all Eigen vectors by their corresponding eigenvalues. Figure 4 shows the top 10 eigenvectors from our training data.

Using concepts from PCA, the images can be reconstructed with a small of number of principal vectors. Figure 5 shows the faces reconstructed by different number of eigenvectors, and we observed that approximately 400 or higher eigenvectors gets us a good reconstruction of the original face image. So in our project, we applied the top 400 eigenvectors as the representation of images. Due to computational reasons during training, only 1400 training face images (200 per class) were used.

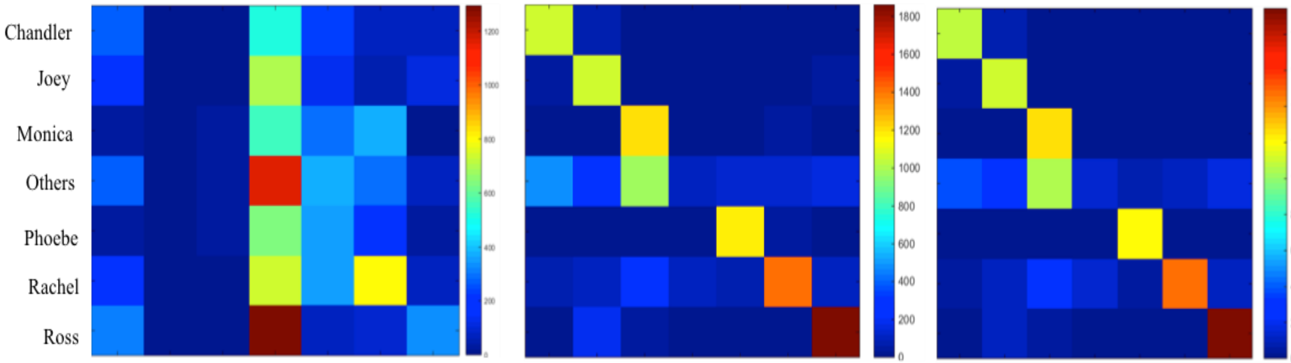
With the low dimension representations of images, the classification algorithm can be described as:

(1) Projecting all training samples into the PCA subspace.
(2) Projecting the query face into the PCA subspace.
(3) Classify the query face by comparing its position in the PCA subspace with the positions of training samples, using nearest neighbor method.

4.2 Convolutional neural nets with SVM

For this multiclass classification task, ideally we would have wanted to train the model using thousands of images from the web. Due to lack of images of all the six celebrities (Chandler, Joey, Rachel, Monica, Ross and Phoebe) in the preexisting dataset, we created our own dataset. After running the Viola-Jones face detector (as described in section 3) on episodes from season one to eight, we collected ~300,000 face images. We then manually labeled a subset of these (approximately 15,000) images to train the model. For the 'others' class, we took 400 images of 4 celebrities from the preexisting dataset, different from the rest of the six celebrities. After building this training dataset, which consisted of approximately 2000 images per class, these were then used to train a combination of an off the shelf convolutional neural networks (CNN) and a support vector machine (SVM) [12].

Previous works [13] have shown that off-the shelf trained CNNs, which are trained on quite different datasets such as ImageNet can give good performance and should be treated as baseline for any recognition task. Hence, a very deep (19-layer) off the shelf model [14], which was trained for days on clusters of GPU, was used for our classification problem. This could be visualized as for a single face image, reducing the dimensionality of it such that the resulting feature vector is a very good representation of the original image, quite similar to concepts of PCA. This resulting feature vector could be extracted from different layers, however the first



Classification Technique	Chandler	Joey	Monica	Others	Phoebe	Rachel	Ross	Total Accuracy
Eigen Faces +NN	24.14	1.68	1.62	52.68	28.75	37.25	15.62	26.16
CNN + SVM	91.21	89.31	93.05	4.56	94.24	66.59	87.37	69.35
CNNaug + SVM	**92.32**	**92.76**	**97.06**	**6.31**	**95.68**	**68.6**	**91.17**	**71.87**

Figure 6: Confusion matrices and accuracy for the three models when tested on ~13,000 frames from Friends season nine: (a) Eigen faces with nearest neighbor classifier, (b) CNN with linear SVM and (c) CNNaug with linear SVM (from left to right)

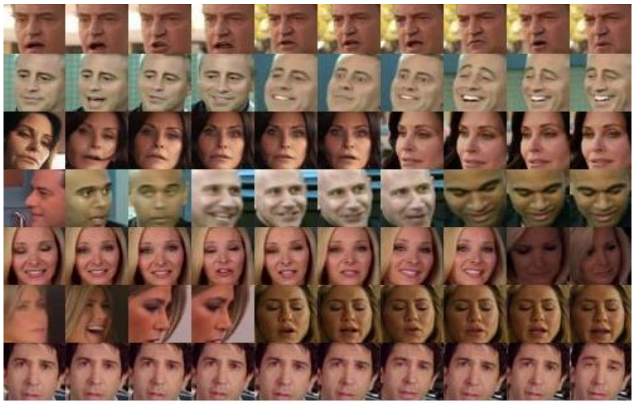


Figure 7: Images of faces of all the six celebrities from video of Friends movie which show distinct variation in appearance due to ageing when compared with Figure 10.

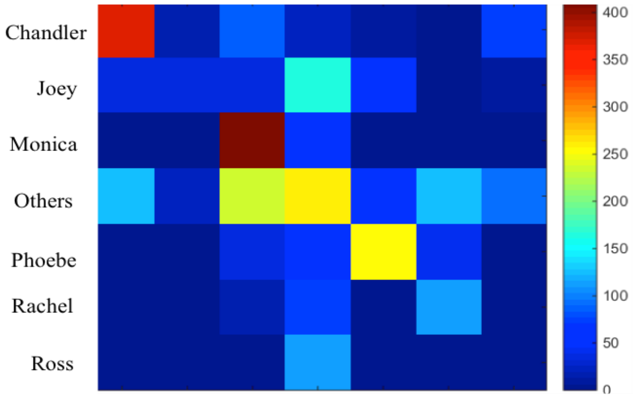
fully connected layer gave the best results. Further, these feature vectors were then used as the input to a linear SVM. Being a multi-class classification problem, the one vs rest approach with a cost penalty (for misclassification) of 3 gave the best performance.

5. Results and Discussion

Results of all three trained models, namely Eigen faces with nearest neighbour classifier, CNN with linear SVM and CNNaug with linear SVM when tested on ~13000 face images from Friends season 9 are summarized in Figure 6. We observed an overall accuracy of 26.16 percent using Eigen faces. The accuracy is improved to nearly 69.35 percent with just using an off the shelf CNN and linear SVM. Further training the same model with augmented data (by flipping the images) earned an extra two percent of accuracy.

One of the possible reasons for low performance of Eigen faces with nearest neighbour classifier could be due to training using only 200 faces per class. As seen from the confusion matrix shown in Figure 6 almost a large number of the faces were falsely predicted as the class ‘others’, which could be partially due to fact that when a query face is projected on the PCA subspace, due to lack of enough training data, its easily misclassified by computing the label from the nearest neighbour. One possible way to overcome that could be using a weighted combination or using K nearest neighboring and taking majority vote. With using CNN and linear SVM we observed an improved accuracy even in individual classes, which is clearly depicted from the diagonal entries in the confusion matrix except for the class ‘others’. One possible reason for the misclassification of class ‘others’ could be that during training, images from only 6 celebrities were used to train the model. As a result during the testing phase face images, it is highly unlikely that the class others will be classified to match these six celebrities. A similar observation was observed when the training dataset was flipped (mirror image) to train the same model, CNN with linear classifier (CNNaug with linear classifier). There was only a two percent increase in the overall accuracy.

We also tested the best model (CNNaug with linear classifier) on video of Friends movie, which was released in 2014. This video is quite different from season 9 as it was made approximately 10 years later than the Friends TV show as shown in Figure 7. We observed an overall accuracy of 47.35%, which is roughly a 20% decrease from



Chandler	Joey	Monica	Others	Phoebe	Rachel	Ross	Overall
63.34	10.46	85.62	28.03	64.39	56.06	0	47.35

Figure 8: Confusion matrix and accuracy (in percentage) of CNNaug with linear SVM on Friends movie video dataset for all the seven classes.

accuracy on seasons 9 dataset as shown in Figure 8. These observations are quite similar to those in a previously published research paper [15]. This is primarily due to the fact that with aging appearances of the faces of each celebrity changes a lot.

6. Conclusion and Future work

To conclude, this paper presents an approach to identify character names of faces detected from video clips, using off the shelf CNN with a linear SVM training model. Experiment shows that accuracy of nearly 70% could be achieved on videos from Friends season nine. Further we also showed the observation that with aging the accuracy dropped to about 48%, which could be reasoned that aging tends to change the appearances of the faces.

In the future, we would like to perform various clustering techniques like agglomerative clustering, EM clustering on the testing dataset to extract face tracks and thus improve the accuracy even further. Another possible research direction from here could be of building 3D face reconstructed models. Given we have nearly 5000 face images of each celebrity captured from different poses, it is highly likely that a 3D face model can be made for each character, which could then be used for tasks like recognition etc.

References

1. Hulu Face Match. http://www.hulu.com/labs/tagging
2. T. Kanade, "Picture Processing System by Computer Complex and Recognition of Human Faces," Kyoto University, Japan, PhD. Thesis 1973
3. L. Sirovich and M. Kirby, "Low-dimensional Procedure for the Characterization of Human Faces," Journal of the Optical Society of America A: Optics, Image Science, and Vision, Vol.4, pp.519-524, 1987
4. Y. Moses, Y. Adini, and S. Ullman, "Face recognition: the problem of compensating for changes in illumination direction," in European Conf. Computer Vision, 1994, pp.286-296.
5. F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision. Sarasota, FL, USA, 1994, pp.138-142.
6. Lawrence, Steve, et al. "Face recognition: A convolutional neural-network approach." Neural Networks, IEEE Transactions on 8.1 (1997): 98-113.
7. Satoh, Shin'ichi, Yuichi Nakamura, and Takeo Kanade. "Name-it: Naming and detecting faces in news videos." IEEE Multimedia 6.1 (1999): 22-35.
8. A. Howell and H. Buxton, "Towards unconstrained face recognition from image sequences," in Proceedings of the Second IEEE International Conference on Automatic Face and Gesture Recognition, 1996, pp.224-229.
9. Viola, Paul, and Michael Jones. "Robust real-time object detection." International Journal of Computer Vision 4 (2001): 34-47.
10. Turk, Matthew, and Alex Pentland. "Eigenfaces for recognition." Journal of cognitive neuroscience 3.1 (1991): 71-86.
11. LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." The handbook of brain theory and neural networks 3361 (1995): 310.
12.Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers." Neural processing letters 9.3 (1999): 293-300.
13. Razavian, Ali Sharif, et al. "CNN Features off-the-shelf: an Astounding Baseline for Recognition." Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on. IEEE, 2014.
14. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
15. Ramanan, Deva, Simon Baker, and Sham Kakade. "Leveraging archival video for building face datasets." Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE, 2007.

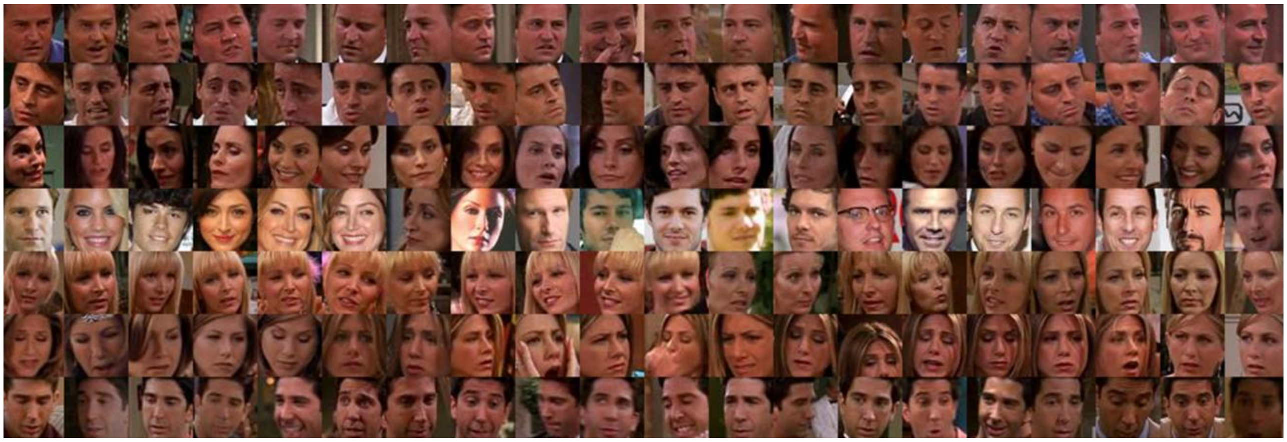


Figure 9: Training Data: Images of faces of all the six celebrities extracted using the Viola-Jones face detector on frames from Friends season one to eight. The fourth row from the top contains face images of others using pre-existing dataset.

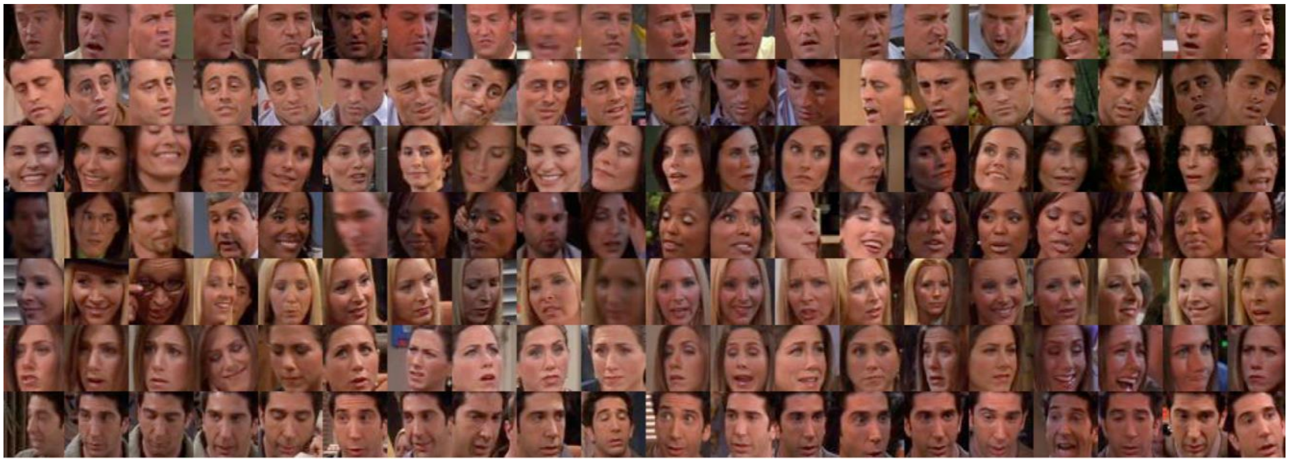


Figure 10: Testing Data: Images of faces of all the six celebrities and others extracted using the Viola-Jones face detector on frames from Friends season nine.