CSCI-B 457 Final Project Report: Celebrity Facial Recognition

Sam Greenfield
Indiana University
spgreenf@iu.edu

1. Introduction

Facial Recognition is a subset of image classification, one of the core problems in computer vision. While the use of Deep Learning and Convolutional Neural Networks is becoming more common in the field of face classification [2, 7], classical techniques can still prove useful. This project explores whether the Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM) pipeline can be used to accurately distinguish between faces.

The FaceScrub dataset [4] is used to build a dataset of images labeled by the name of the person in the image. Then, the images are normalized and augmented before being split into training and testing images. HOG features are extracted from each, and SVMs are trained on the training images and labels. Finally, the classifiers are tested by feeding each a testing image in a one-vs-rest format, and comparing the predicted label to the actual one.

A basic goal for this project is to have reasonable accuracy, such that the product is useful in recognition tasks. It should also be reasonably timely, not taking hours and hours to classify images. Secondary goals include ease of use and adaptability to other datasets. This project was made in a Jupyter notebook utilizing Python and Markdown, and downloaded images are excluded from the source code to reduce file size.

2. Dataset Construction

2.1. FaceScrub Dataset

FaceScrub is a 2014 dataset comprised of text files listing entries of URLs to images and the names of the celebrity in the images. It contains images of 530 people and links to 106,863 images [4]. This dataset was made with the intention to aid in computer vision and facial recognition research, and is relevant to the task of this project because it provides labeled images of celebrities, which can be used to train a classification model.

2.2. Class Selection

The first step of the project is class selection. To keep a reasonable scope, only 20 of the 530 classes are used. To determine which classes to use, we identify the top 10 actors and top 10 actresses by count of entries in the dataset. First, a list of entries is made. Next, a Counter from the collections library is used. The top classes by count are tracked as the names of celebrities to download images of.

2.3. Image Downloading and Verification

Since FaceScrub comes as a list of entries including URLs, the images must be downloaded before training a classifier on them. To do this, we use the requests package to save the file to a variable. Then, we ensure that the image is as expected by comparing the sha256 hash provided in the dataset with the hash of the file variable. If the hashes match, we know the image is the same as intended in the original dataset. Otherwise, the image has likely either changed or been removed from the provided URL, and it is thrown out. We end up getting rid of a lot of images this way since the dataset is over a decade old and many URLs have changed.

As the images are downloaded, they are sorted into a folder labeling them as images of actors or actresses, and into a subfolder with the person's name as the folder name. This step is vital, as the folder name is used as the label in classification. After downloading each image, a random image of each person is displayed to give the user a brief overview of the classes which will be classified in future steps (1).

3. Preprocessing and Feature Extraction

3.1. Face Cropping and Normalization

In order to increase the accuracy of the facial recognition, we first crop the image to where there is a face. To do this, the OpenCV Frontal Face Haar Cascade classifier is used to detect the region where there is a face. Next, we take only the crop of the original image where the face is.

To normalize the images before descriptor processing, we take the cropped region and resize it to 256x256. This seems to strike a reasonable balance of detail and efficiency, as a larger image would have a much larger HOG descriptor which would take a much longer period to compute, but

Figure 1. After downloading and image verification, the top 10 actors and top 10 actresses are shown with their names as labels. This gives the user a reference point for which celebrities are used as classes in the classification step.

a smaller image would lose facial details which are necessary to differentiate classes. The image is also converted to grayscale, as this is the format the HOG feature extraction takes as input.

3.2. Augmentation by Rotation

The final change to the images before feature extraction is rotation. Since the downloaded data after validation contains only between 36 and 100 images per class, there are 1502 total images across the 20 classes. With a 30% test split, only about 1050 images are used for training 20 classes. To increase the number of test images, and, therefore, the accuracy of the classifiers, we augment each image by rotating it a little bit.

For each grayscale crop in the training group, the rotations of the image -10°, 0°, and 10°are appended to the test dataset under the same label as the original crop. This rotation augmentation is sound as faces in test and training images may be at different angles, meaning that a good classifier will need to take into account different angles of faces. Unlike with text classification, where rotation may distort a model's ability to classify (for example, rotation of the digit '6' may lead to misclassification of a '9' [8]), this model performs better with the extended training set. The cropping, normalization, and rotation are visualized in 2.

3.3. HOG Feature Extraction

This step of the pipeline is where the data is split between training and testing data. This is done by shuffling the images and storing the first 30% as testing and the rest as training. Only training images are rotated as discussed

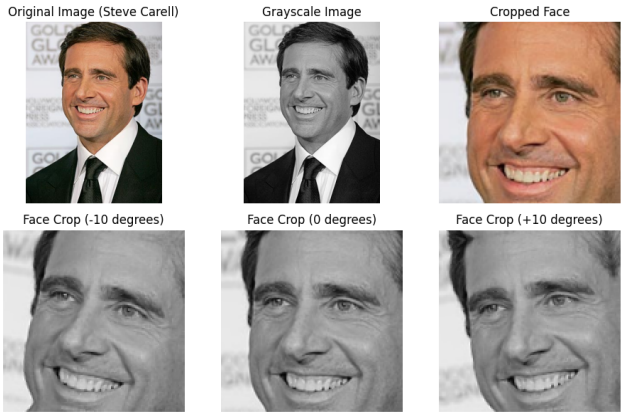

Figure 2. Cropping, resizing, grayscale conversion, and rotation of an example image of Steve Carell.

previously. However, every image is run through the Haar Cascade to crop to the face. Since the data are shuffled during this step, there will be a little bit of variability in the trained classifiers and therefore results.

The Histograms of Oriented Gradients feature extraction method relies on consistencies between interesting points of an image. This is akin to other descriptors, such as Scale-invariant feature transform (SIFT). The HOG extraction used in this project takes in an image (cropped, scaled, color-converted, rotated, like the second row in 2), and begins by computing the horizontal and vertical gradient images using OpenCV's Sobel function. Next, the magnitude and angle of the gradient are each computed as 2-dimensional NumPy arrays.

Next, for each non-overlapping 8x8 region of the image, called a cell, a 9-bin histogram is computed with values based on the magnitude of pixels within the bin's range of angles. Then, for each 2x2 overlapping block of cells, the histograms are flattened, concatenated, and normalized (with respect to the block) into a vector representing the block. The vectors for each block are concatenated into one descriptor vector, which now represents the image in the eye of the classifier. The HOG descriptor is a good representation of the image, as it takes into account local spacial information [1] without being dependent on exact image matching.

With this understanding of HOG descriptors, the safety of rotation may be more intuitive. Since the HOG descriptor bases gradients on only 9 blocks of angles, a $\pm 10°$difference in angle may flood information into slightly different bins, which can lead to small differences in the final descriptor. These small changes help ensure that important features are not disregarded because of edge cases (for example, if the crook of a subject's eye, a unique feature, is split between two cells or blocks, there may be some information lost).

Since 256x256 images are the input of the HOG feature extraction, there are 32x32 cells of 8x8 pixels. Since the block is 2x2 cells overlapping, there are 31x31 blocks. Each block has 4 cells, which each have 9 values (one per bin of histogram). So, the total size of each HOG descriptor vector is $31 * 31 * 4 * 9 = 34,596$.

4. One-vs-Rest Linear Classifier

We train many one-vs-rest linear classifiers to build a multi-class classification model. So, when training, each class gets its own classifier [3]. To implement this, we loop through each label in the training set, and create a new list of labels which, instead of labeling images as their class name (i.e. "Which celebrity is this?"), we label images according to whether or not they are the current label (i.e. "Is this image [Steve Carell, Kristen Chenoweth, Alec Baldwin, etc.]?").

The classifier used is the scikit-learn LinearSVC (support vector classifier). It runs with up to 5,000 iterations to properly distinguish between HOG descriptors which are and are not from the label. After a classifier is built for each label, it is saved to a dictionary of all classifiers, indexed by the label it classifies.

In order to make classifications, we loop through each classifier and its respective label and feed the list of HOG feature vectors into the classifier's decision function, which returns a list of confidence scores for each image. Then, for each image, we assign to the image the label of the classifier which gave it the highest confidence score.

5. Results and Evaluation

The method for evaluation in this project comes from scikit-learn's classification report (3) and confusion matrix (4). These help identify a few factors in evaluation. First, some terms must be defined. For a class A and any other class B, and for an image I:

- A prediction is a true positive (TP) if $I \in A$ and the model also predicts $I \in A$. This is good, as it means that the model correctly classified the test image.
- A prediction is a false negative (FN) if $I \in A$ and the model predicts $I \in B$. This is bad because it means that a classifier was supposed to identify the image as positive, but instead thought it was negative.
- A prediction is a false positive (FP) if $I \in B$ and the model predicts $I \in A$. This is bad because it means that a classifier was supposed to identify the image as negative, but instead thought it was positive.

The success metrics are as follows [5, 6], with values of 1 being perfect and 0 being poor:

- Precision: The rate at which positive predictions are true positives.

$$P = \frac{TP}{TP + FP} \tag{1}$$

- Recall: The rate at which positive images were detected as positives.

$$R = \frac{TP}{TP + FN} \tag{2}$$

- Accuracy: The rate at which predictions (positive or negative) were correct.

$$A = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

5.1. First Try: Disappointing Results

Before refining the model, a classification cycle was run without face cropping or rotation, with the only normalization being resizing to 256x256 and conversion to grayscale. Classifiers were given only 1,000 iterations to converge. It took 43 seconds to develop the HOG features for the dataset, and 3 minutes, 58 seconds to build classifiers. In this run, the model had very low accuracy. This is likely because the images from the dataset have to be resized to the standard. This means that faces are distorted, making it difficult for the model to converge on a meaningful classification. With a weighted average of 0.16 precision and 0.17 recall, and an overall accuracy of 0.17, adjustments had to be made to improve the model.

5.2. Subsequent Tries: Lots of Progress

For the next iteration, the Haar Cascade face crop was introduced for the training data, but rotations had not been included. This method had significantly better results. The HOG features were computed in 1 minute, 40 seconds, and the classifiers were trained in 1 minute, 53 seconds. The model also had a huge jump in success metrics with 0.79 precision, 0.77 recall, and 0.77 accuracy. These results are meaningful, but could still be better.

Next, the rotations were introduced. Since the training dataset was three times as large, the HOG feature extraction took much longer (2 minutes, 33 seconds). Classifier training took significantly longer as well (8 minutes, 28 seconds). However, the success metrics also increased significantly to 0.82 precision, 0.79 recall, and 0.80 accuracy. However, throughout each of these runs, the SVCs were limited by the number of iterations in their training, meaning that some did not converge.

For the final product, the SVCs were given 5,000 iterations to converge and the margin of error for the SVC was

	precision	recall	f1-score	support
Alec Baldwin	0.89	0.80	0.84	20
America Ferrera	0.87	0.90	0.88	29
Ben Stiller	0.87	0.81	0.84	16
Bill Hader	0.74	0.70	0.72	20
Christina Applegate	0.67	0.74	0.70	27
Colin Farrell	0.69	0.96	0.80	23
Colin Firth	0.61	0.76	0.68	25
Courteney Cox	0.74	0.83	0.78	30
Debra Messing	0.81	0.93	0.86	27
Fran Drescher	0.85	0.96	0.90	24
George Clooney	0.84	0.76	0.80	21
Hugh Grant	0.86	0.63	0.73	19
Kristin Chenoweth	0.88	0.72	0.79	29
Matt Damon	0.83	0.79	0.81	24
Nadia Bjorlin	0.83	0.50	0.62	10
Neve Campbell	0.79	0.77	0.78	30
Patricia Arquette	0.80	0.60	0.69	20
Roseanne Barr	1.00	0.86	0.92	14
Simon Pegg	0.94	0.80	0.86	20
Steve Carell	0.83	0.91	0.87	22
accuracy			0.80	450
macro avg	0.82	0.79	0.79	450
weighted avg	0.81	0.80	0.80	450

Figure 3. Screenshot of classification report for classification testing with face cropping, rotation, and maximum 5,000 iterations for SVC training.

```
[[16  0  0  0  0  1  0  0  0  0  1  1  0  0  0  1  0  0  0  0]
 [ 0 26  0  0  0  0  2  0  1  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 13  1  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0]
 [ 0  0  1 14  0  0  2  1  0  0  0  0  0  0  0  0  1  0  1  0]
 [ 0  0  0  0 20  2  0  0  0  1  0  0  1  0  1  0  2  0  0  0]
 [ 0  0  0  0  0 22  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 1  0  0  1  1  0 19  0  1  0  0  0  0  0  0  0  0  0  0  2]
 [ 0  0  0  0  1  0  1 25  2  0  0  0  0  1  0  0  0  0  0  0]
 [ 0  0  0  0  0  1  0  1 25  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 23  0  0  1  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  2  1  0  0  0 16  0  0  0  0  1  0  0  0  0]
 [ 1  0  0  0  0  0  0  1  1  0  0 12  0  1  0  1  0  0  0  2]
 [ 0  0  0  0  1  1  0  4  0  2  0  0 21  0  0  0  0  0  0  0]
 [ 0  0  0  0  1  0  1  0  1  0  1  0  0 19  0  1  0  0  0  0]
 [ 0  0  0  0  3  0  1  1  0  0  0  0  0  0  5  0  0  0  0  0]
 [ 0  1  0  0  1  1  2  0  0  1  0  1  0  0  0 23  0  0  0  0]
 [ 0  1  1  0  1  0  1  0  0  0  0  0  1  1  0  2 12  0  0  0]
 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 12  0  0]
 [ 0  0  0  0  0  2  1  0  0  0  0  0  0  1  0  0  0  0 16  0]
 [ 0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 20]]
```

Figure 4. Screenshot of confusion matrix for classification testing with face cropping, rotation, and maximum 5,000 iterations for SVC training.

increased from 0.0001 to 0.01. In this case, each classifier did converge, but results were largely similar to those of the previous run. HOG feature computation was unaffected, but because of the reduced margin of error, classifiers were trained in just 49 seconds. Just the same as the previous run, there was 0.82 precision, 0.79 recall, and 0.80 accuracy. The confusion matrix and classification report for this run are shown in 3 and 4, respectively. Through different runs with the same configuration, the results were fairly consistent with maximum 0.82 and minimum 0.79 accuracy.

5.3. Error Analysis

As expected, the classification of the HOG descriptor and SVC pipeline is imperfect. It may be helpful to visualize some of the errors made by the model in order to identify potential causes of error. For the final run mentioned in 5.2, we will look into the classification of images in a single, reasonably representative, class: Steve Carell. 5 shows that for this run, there were 48 original training images of Carell, expanded into 144 with rotations. There were 22 images of Carell in the test set. 20 were correctly classified (6), there were two false negatives (7), and four false positives (8). We will look into both the false negative and false positive cases in detail.

The first image (left to right) of Steve Carell in 7 was not detected by the model. This image is of Carell showing only the right side of his face with much of the left side of his face obscured. Looking at the training set, the only images with him showing almost exclusively the right side of his face are the images in the middle of row 3 and in the middle of row 13. From this, it is inferred that the cause of the false negative is a lack of relevant training data for this test image. The second false negative image is of Carell sipping from a mug, obstructing his mouth and chin. Looking at the training set, there are no images the model trained on which include an obstruction to the bottom half of Carell's face. Therefore, it is reasonable to assume the model misclassified this image because it had not been trained on any images with a similar obstruction.

It is more difficult to attempt to identify the reasoning for the false positives in 8 since they are (to the human eye) much less similar to the training images of Carell than the false negatives. However, one can make reasonable inferences for which similar traits between false and true positives may have led to the misclassification. For example, the second image in 8 is of a white man wearing dark colored glasses, similar to many of the training images of Carell. The third image is of a white man with dark hair and shadows around his eyes, similar to training images of Carell (i.e. middle of first row, right side of fifth row, and left side of tenth row). These may be the reasons that the HOG descriptors for the false positives were classified as being of Carell.

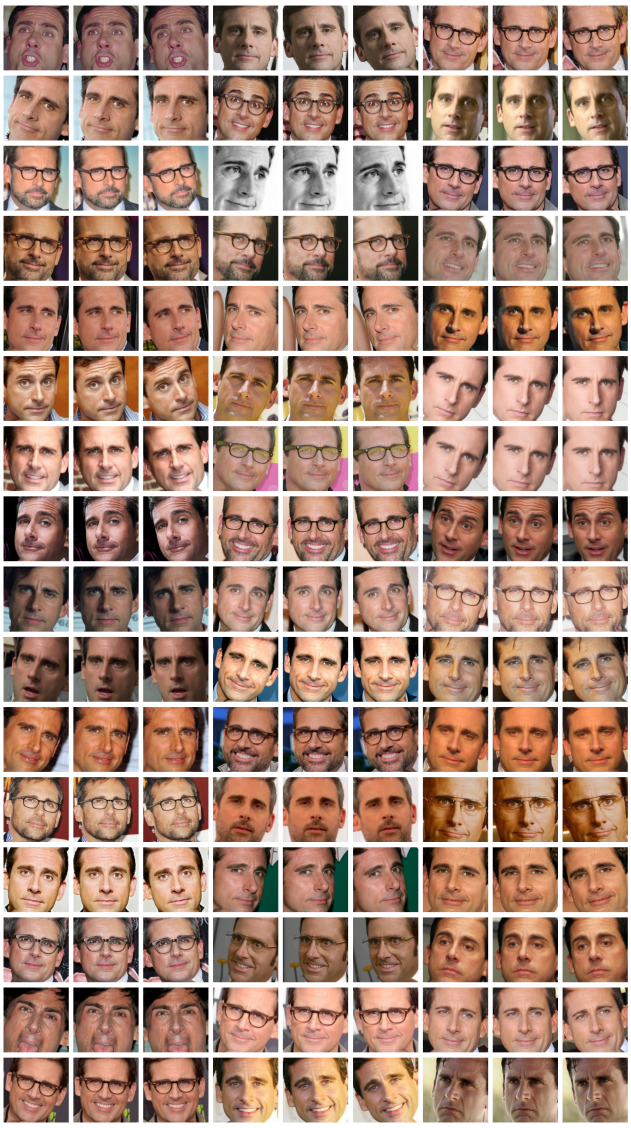

Figure 5. Subset of training data for class "Steve Carell", cropped to face and expanded by performing rotations.

Figure 6. True positive detections of Steve Carell (images of Carell which were correctly identified by the model).

Figure 7. False negative detetions of Steve Carell (images of Carell which were identified as another class).

Figure 8. False positive detections of Steve Carell (images not of Carell, which the model mistakenly identified as Carell).

6. Discussion and Conclusion

6.1. Strengths

HOG and SVM proved to be surprisingly effective at identity classification on this real-world dataset of celebrities. While the FaceScrub dataset is clean in its labels, it includes many images which include features besides the person's face. This makes classification more difficult, but the preprocessing of images, including the Haar Cascade crop and rotation contribute significantly to the model's ability to predict the identity of the person in an image. So, this method is shown to still have very acceptable accuracy. With precision around 80%, this model makes a convincing argument for classification without deep learning or neural networks. The model is also valuable because it is lightweight and reproducible. The total program is fairly small, with most of the model's size coming from the data it generates in the form of HOG descriptors and SVC objects. While this method is not deterministic, relying on the randomness in the data split and SVC training, the testing accuracy falls within a reasonable 3% range between runs, demonstrating fairly good stability. Finally, the training period was cut down significantly to under a minute with the same high accuracy.

6.2. Limitations

This method has its merits, but is imperfect. It is particularly sensitive to unexpected poses and occlusion, as discussed in 5.3. In addition, SVMs rely on linear classification, so the model is not good at recognizing complexities in the difference between facial classes. It may also struggle in real-world conditions not accounted for, such as variations in image noise. With a larger dataset and more classes, the runtime of the program would increase significantly, so testing is necessary. While extracting features from 256x256 images increased descriptiveness, it also resulted in a 34,596 dimensional vector, which made training slower than necessary, although acceptable. Finally, the FaceScrub dataset did not prove to be very racially diverse. Of the top 10 actors and actresses selected, almost all were White. This is not a good representation of the population

of celebrities. It would be good to use a larger subset of the FaceScrub dataset which includes people of various races and ethnicities.

6.3. Future Progress

In the future, it would be useful and interesting to run the classification system on a host of different configurations, varying number of classes, images per class, train/test split, number of rotations, HOG input size, HOG cell and block size, and maximum training iterations on SVC. From there, one could optimize the system to get a good mix of accuracy and computational efficiency. Also, since so many images from the FaceScrub dataset were not available to the system because of invalid URLs, it would be interesting to test this on a newer, larger dataset with more images per class and more diversity in the classes tested.

6.4. Conclusion

This project demonstrates the ability of a HOG and SVM pipeline to achieve high accuracy in facial detection of celebrities. Through various iterations of the HOG extraction and SVM, high-quality results were achieved. Data augmentation via face cropping and rotation proved to be incredibly important in increasing the reliability of the system.

In the development of this project, I learned how to design an end-to-end pipeline for a computer vision product. I also gained a deeper understanding of the HOG feature extraction method, learned about one-vs-rest classification using linear SVMs, and got a taste of error analysis, making predictions about the biases my system may have in prediction which led to misclassification.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. 2

[2] Hossam Mahmoud Elian, Gamal M Dousoky, and Ali Hafez. Developing deep learning based facial recognition technique. *Journal of Advanced Engineering Trends*, 44(1):214–220, 2025. 1

[3] Jin-Hyuk Hong and Sung-Bae Cho. A probabilistic multi-class strategy of one-vs.-rest support vector machines for cancer classification. *Neurocomputing*, 71(16):3275–3281, 2008. Advances in Neural Information Processing (ICONIP 2006) / Brazilian Symposium on Neural Networks (SBRN 2006). 3

[4] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 343–347, 2014. 1

[5] Scikit Learn Documentation. 3.4. Metrics and scoring: quantifying the quality of predictions. 3

[6] Scikit Learn Documentation. Precision-Recall. 3

[7] JiangRong Shi and Li Zhao. Face recognition system based on capsule networks. *International Journal of Advanced Network, Monitoring and Controls*, 9(1):22–31, 2024. 1

[8] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60, July 2019. 2