Human Face Classification Using Machine Learning: A Comparative Analysis

Noam Cohen & Yuval Vogdan

March 2025

Abstract

This project explores various machine learning approaches for human face classification, implementing and comparing multiple algorithms on a modified version of the Labeled Faces in the Wild (LFW) dataset. Originally designed for face recognition, the dataset lacked pre-defined classification labels and required significant adjustments to repurpose it for face classification tasks. We evaluate the effectiveness of traditional machine learning methods including Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Decision Trees, alongside deep learning approaches using Convolutional Neural Networks (CNN) with a specific focus on transfer learning with the VGG16 architecture. The study compares preprocessing techniques, feature extraction methods, and classification algorithms to determine optimal approaches for classifying human faces. Our findings demonstrate that transfer learning with VGG16 achieves the highest accuracy, while also examining the trade-offs between model complexity, computational efficiency, and recognition performance across all implemented methods. Furthermore, we investigate how different minimum thresholds for photos per person (9 and 15) affect model performance, addressing one of the key challenges of the LFW dataset where many individuals have only a single photo.

Contents

1 Introduction

Face classification technology has evolved significantly in recent years, with applications spanning security systems, social media, and human-computer interaction. This project implements and evaluates multiple machine learning approaches for face classification using a modified version of the Labeled Faces in the Wild (LFW) dataset.

Our research explores how different classifiers address the inherent challenges of face classification, with a particular emphasis on comparing traditional machine learning methods with modern deep learning architectures.

1.1 Research Objectives

Our study aims to address the following key questions:

1. How do different preprocessing techniques affect classification accuracy?
2. What is the impact of dimensionality reduction on classification performance?
3. How do traditional machine learning algorithms compare to deep learning approaches in terms of accuracy and computational efficiency?
4. Can transfer learning with pre-trained models (VGG16) significantly improve classification performance?
5. How does the minimum number of photos per person in the training set affect model performance and generalization?
6. How effective are HOG features for face classification compared to raw pixel values?
7. What is the optimal number of principal components in PCA to balance dimensionality reduction and classification accuracy?
8. How does the combination of HOG features with PCA affect both computational efficiency and classification performance?

1.2 Project Scope

The project encompasses the full machine learning pipeline for face classification, addressing both traditional machine learning and deep learning approaches:

- Data preprocessing and augmentation
  - Face detection and alignment
  - Histogram equalization and normalization
  - Image resizing (essential for CNN-based approaches)
  - Data augmentation (primarily for deep learning models)
- Feature extraction and selection
  - HOG feature extraction (for traditional ML models)
  - PCA dimensionality reduction (for traditional ML models)

  - Raw pixel representation (baseline)
  - Deep feature extraction via VGG16 (for deep learning approaches)

- Model training and optimization
  - Traditional classifiers (SVM, k-NN, Decision Trees, Random Forest, AdaBoost, Logistic Regression)
  - Hyperparameter tuning for traditional models
  - Custom CNN architecture design and training
  - Transfer learning with VGG16 (feature extraction and fine-tuning)

- Analysis of data sampling strategies
  - Impact of minimum photos per person threshold
  - Class imbalance handling
  - Training/testing split optimization

2 Dataset

2.1 Labeled Faces in the Wild (LFW)

The Labeled Faces in the Wild (LFW) dataset contains over 13,000 facial images collected from the web. Each image is labeled with the name of the person pictured, with 1,680 individuals having two or more distinct photos. The dataset consists of JPEG pictures of famous people, with each face centered in the image. The original images are 250 × 250 pixels, detected using the Viola-Jones face detector, with each pixel in RGB channels encoded as a float in the range 0.0 - 255.0.

2.2 Dataset Distribution Challenges

Dataset	**Number of Individuals**	**Total Images**
Original LFW	5,749	13,233
Threshold-9	184	4,558
Threshold-15	96	1,867

Table 1: Comparison of dataset sizes after filtering

One of the primary challenges with the LFW dataset is the imbalanced distribution of images per person. While the dataset contains 1,680 individuals with multiple photos, a significant portion of the subjects have only a single image, making them unsuitable for training classification models that require multiple examples per class.

We filtered the data to include only people with minimum 9 images, that being our base data. The distribution of images per person is illustrated in Figure 2.

This imbalance creates significant challenges for model training and evaluation, as recognition systems typically require multiple examples per person to learn distinguishing features. Models trained on subjects with few samples may have limited generalization capability when faced with new images of the same individuals.

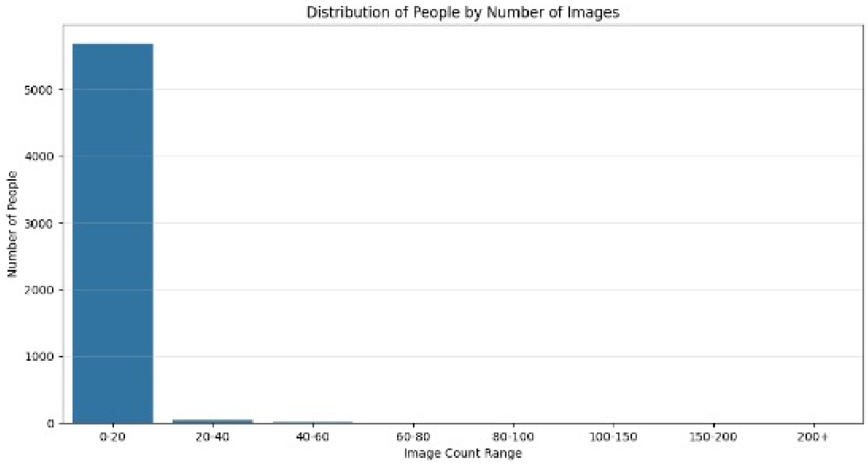


Figure 1: Distribution of photos per person in the original LFW dataset

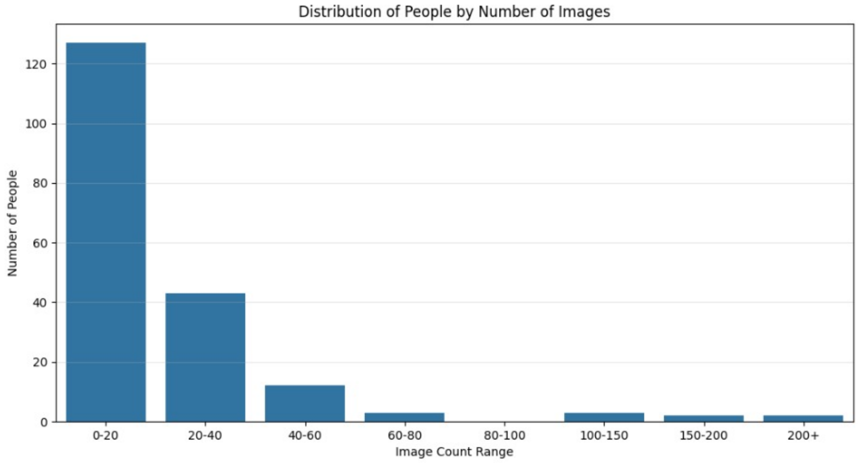


Figure 2: Distribution of images per person in the filtered to 9 minimum LFW dataset. (Approx 130 People with 0-20 images

Additionally, due to the presence of individuals with a large number of images compared to the majority of the dataset, which contains between 0-20 images per person, we decided to balance the data by limiting the number of images per person to a maximum of 40. This approach ensures a more uniform distribution of images across individuals, enhancing the model's ability to generalize effectively in Figure 4.

2.3 Dataset Preparation

To address the challenges of the imbalanced dataset, we experimented with different thresholds for the minimum number of photos per person:

- **Threshold-15 Dataset**: Selected individuals with at least 15 images to ensure sufficient training examples
  - Resulted in a filtered dataset with 96 different individuals
  - More balanced class distribution

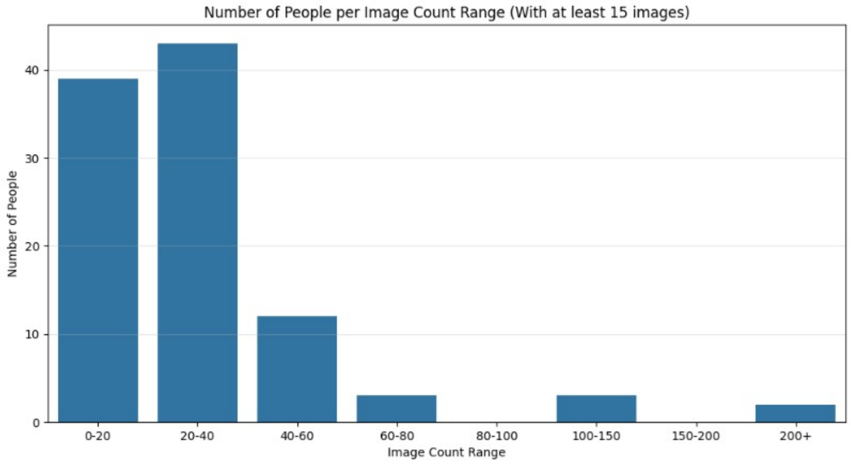


Figure 3: Distribution of images per person in the filtered to 15 minimum LFW dataset (used mainly for the CNN models

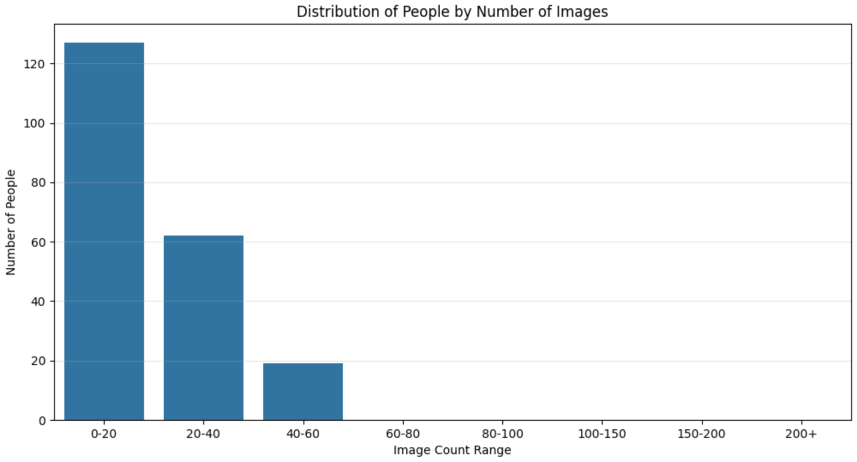


Figure 4: After Setting a maximum of 40 photos per person

  - Reduced overall dataset size

- **Threshold-9 Dataset**: Selected individuals with at least 9 images as a compromise between data quantity and quality
  - Resulted in a filtered dataset with 184 different individuals
  - Larger dataset size but potentially more challenging for models
  - Greater variability in class sample sizes

For both dataset variants, we:

- Split the data into training (80%) and testing (20%) sets
- Applied various preprocessing techniques including normalization, alignment, and resizing
- Ensured stratified sampling to maintain class distribution in both splits

The different thresholds allowed us to examine the trade-off between having more individuals (greater diversity) versus having more photos per individual (better per-class representation).

3 Methodology

Our project implements a comprehensive set of approaches for face classification, organized in a systematic pipeline from preprocessing to evaluation.

3.1 Data Preprocessing

Several preprocessing techniques were applied to enhance the quality of input images:

- Histogram equalization to improve contrast
- Face detection to isolate facial regions
- Image normalization to standardize pixel values
- Resizing images to required dimensions (224×224 for VGG16)

3.2 Feature Extraction Methods

We explored multiple feature extraction techniques:

- Histogram of Oriented Gradients (HOG)
- Principal Component Analysis (PCA)
- Convolutional feature maps (from VGG16)
- Raw pixel values (baseline)

3.3 Implemented Models

Our research evaluated the following models:

3.3.1 Traditional Machine Learning Models

- Support Vector Machines (SVM)
- k-Nearest Neighbors (k-NN) with various distance metrics (1-5)
- Decision Trees with different depth configurations (1-30)
- Logistic Regression
- AdaBoost with decision tree (max depth 1), SVM and logistic regression base.

3.3.2 Deep Learning Models

- Custom Convolutional Neural Network (CNN)
- Transfer Learning with VGG16:
  - Feature extraction (frozen convolutional layers)

  - Fine-tuning (unfreezing later convolutional layers)

Each model was evaluated on both the Threshold-9 and Threshold-15 datasets to analyze the impact of different data sampling strategies.

4 Traditional Machine Learning Approaches

We evaluated multiple traditional machine learning models on our face classification task using a filtered dataset with a minimum of 9 images and maximum 40 images per class, resulting in 184 distinct classes. For each model, we tested four feature configurations:

- Original pixel values (flattened image arrays)
- Histogram of Oriented Gradients (HOG) features
- Principal Component Analysis (PCA) dimensionality reduction
- Combined PCA and HOG features

4.1 Logistic Regression

We implemented logistic regression with an extended iteration limit (max iterations = 1000) to ensure convergence on our high-dimensional face data.

- Original pixels: 28.80% accuracy
- HOG features: 31.84% accuracy
- PCA (1000 components): 27.00% accuracy
- PCA+HOG (500 components): 34.00% accuracy

Logistic regression performed surprisingly well, especially with combined PCA+HOG features, demonstrating that even linear decision boundaries can capture meaningful facial characteristics when appropriate feature extraction is applied. The reduced performance with PCA alone suggests that some discriminative information was lost during dimensionality reduction.

4.2 Support Vector Machine (SVM)

We utilized a linear kernel SVM configuration optimized for margin separation between facial classes.

- Original pixels: 21.85% accuracy
- HOG features: 28.80% accuracy
- PCA (1500 components): 22.00% accuracy
- PCA+HOG (1000 components): 29.09% accuracy

SVM showed moderate performance, with HOG features consistently improving results. The combination of PCA and HOG features produced the best performance, suggesting that both global (PCA) and local (HOG) facial features contribute to classification accuracy. Interestingly, SVM required more PCA components (1000-1500) than other models to achieve optimal results, indicating its ability to leverage higher-dimensional representations.

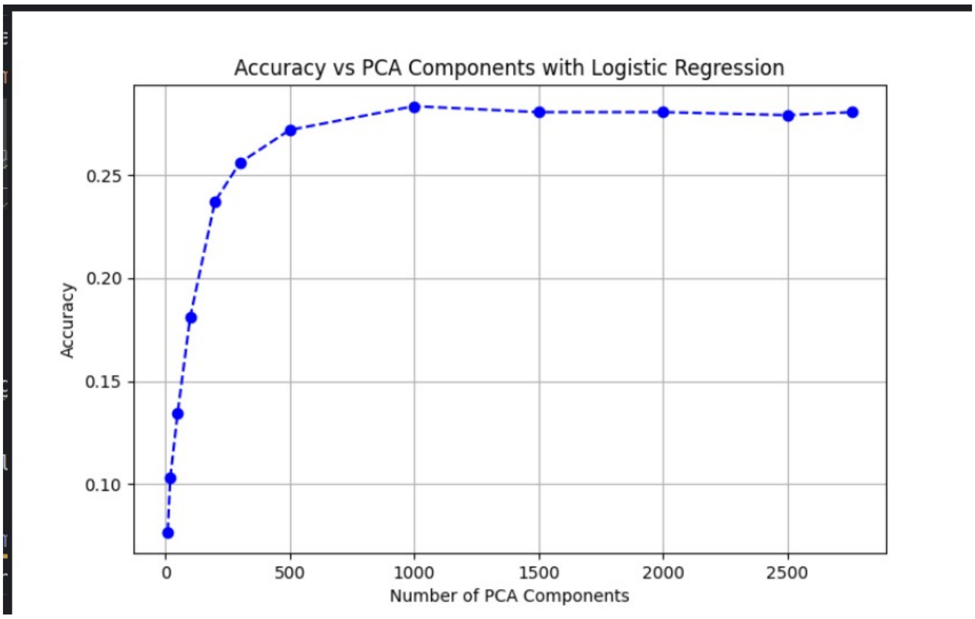


Figure 5: Logistic Regression Model Visualization with PCA

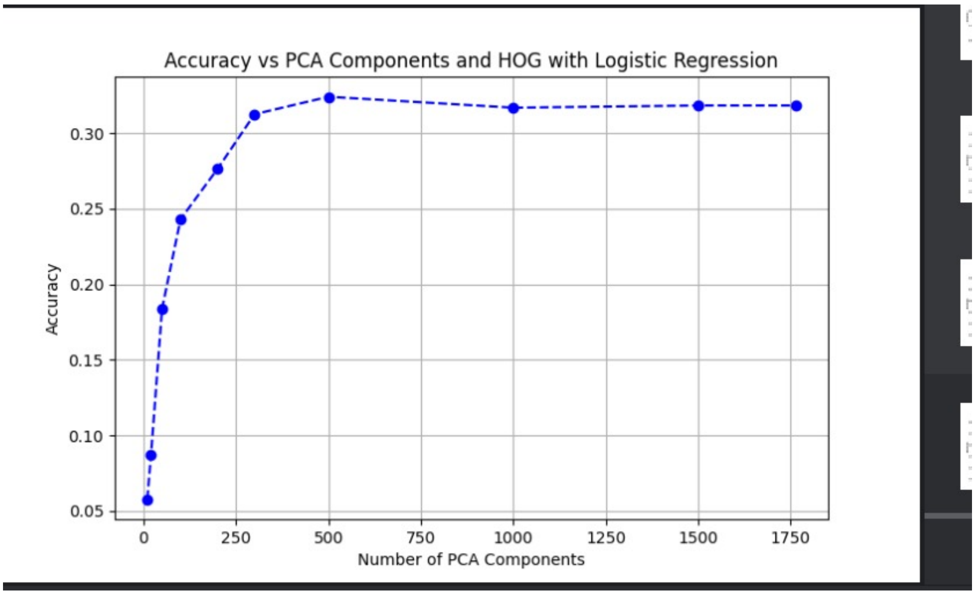


Figure 6: Logistic Regression Model Visualization with PCA + HOG

4.3 Decision Tree

We implemented decision trees with varying maximum depth values to prevent overfitting, testing depths of 1, 5, 10, 15, 20, 30, and unlimited depth.

- Original pixels: 9.70% accuracy (max depth 10)
- HOG features: 4.92% accuracy (max depth 15)
- PCA (50 components): 3.90% accuracy (max depth 10)
- PCA+HOG (50 components): 3.04% accuracy (max depth 10)

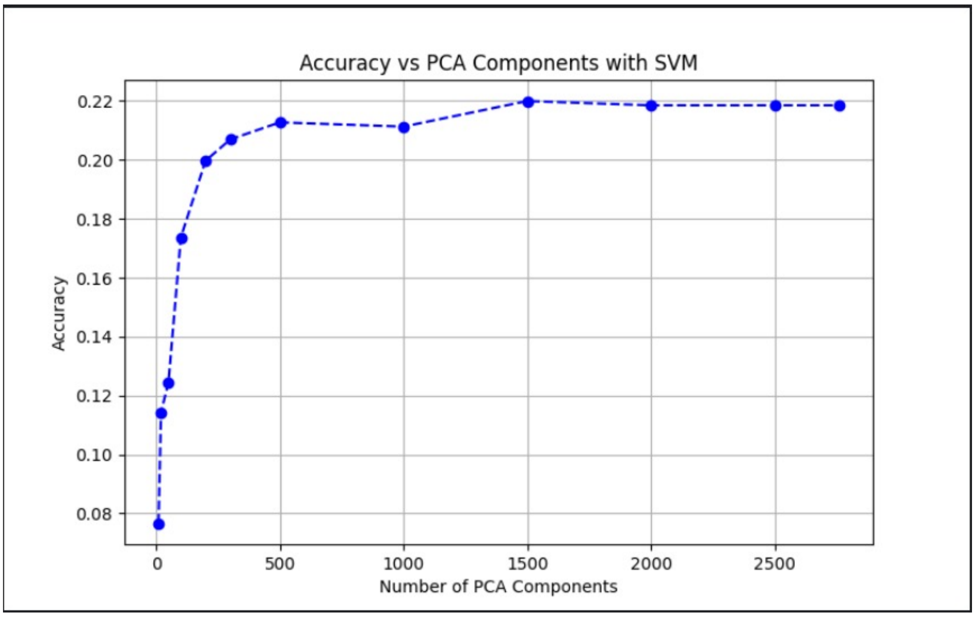


Figure 7: Support Vector Machine Model Visualization with PCA

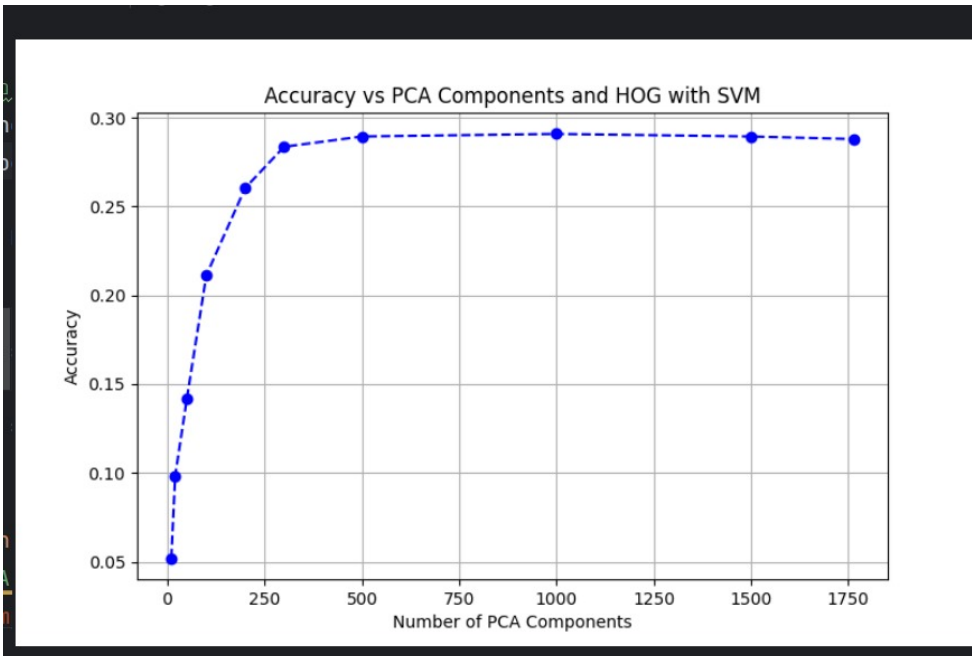
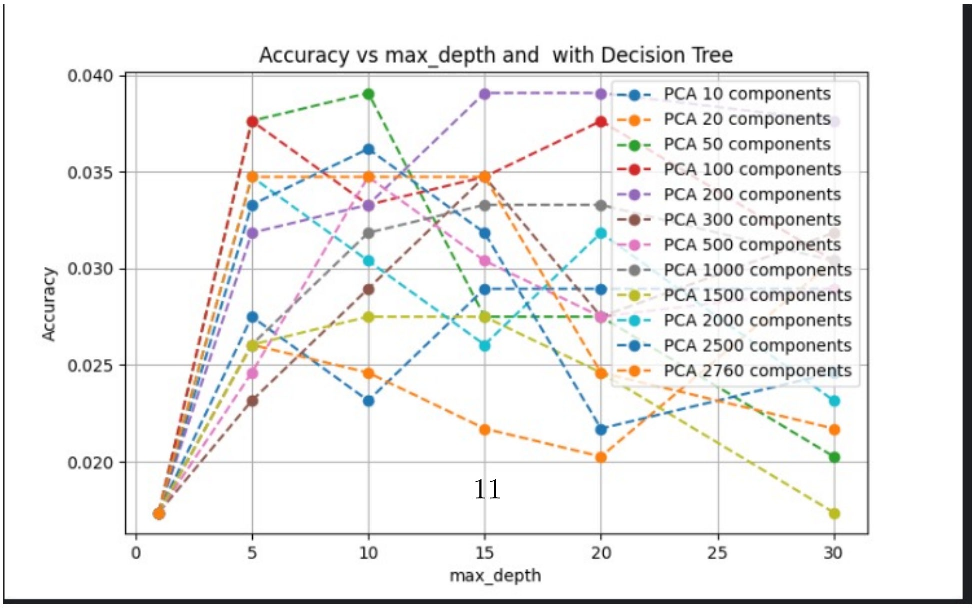


Figure 8: Support Vector Machine Model Visualization with PCA+HOG



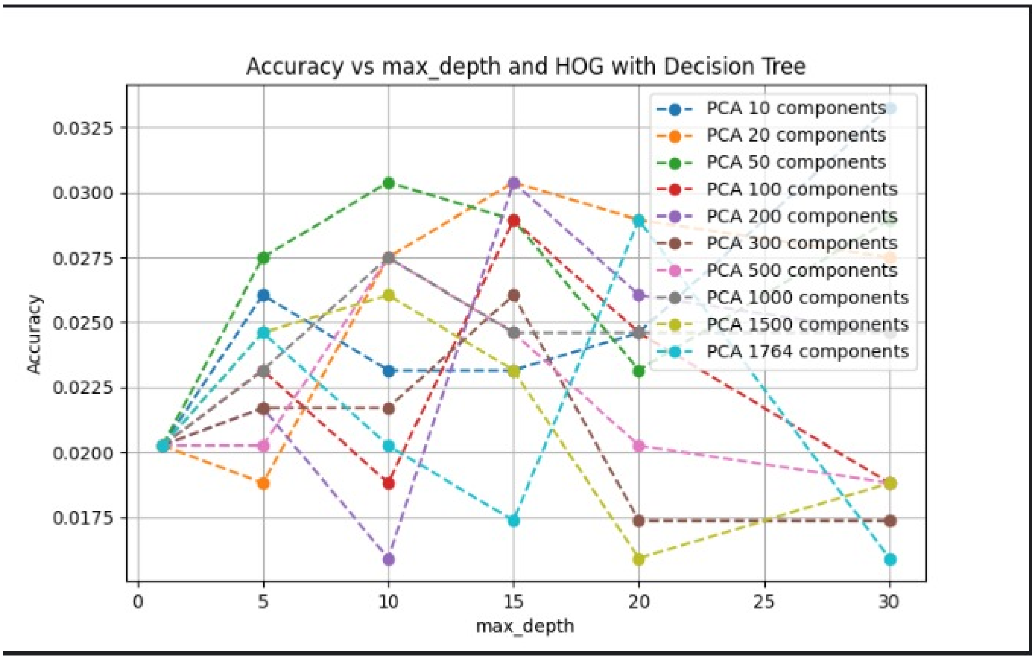


Figure 10: Decision Tree Model Visualization with PCA+HOG

Decision trees performed poorly across all feature configurations, with the best performance on raw pixel values. The significant drop in performance with feature extraction techniques suggests that decision trees struggle with the transformed feature spaces. The optimal depth of 10-15 indicates that excessive tree complexity leads to overfitting in face classification tasks.

4.4 K-Nearest Neighbors (K-NN)

We tested K-NN with multiple K values (1-5) to find the optimal neighborhood size for classification.

- Original pixels: 8.80% accuracy (K=1)
- HOG features: 10.67% accuracy (K=1)
- PCA (100 components): 10.27% accuracy (K=1)
- PCA+HOG (500 components): 16.64% accuracy (K=1)

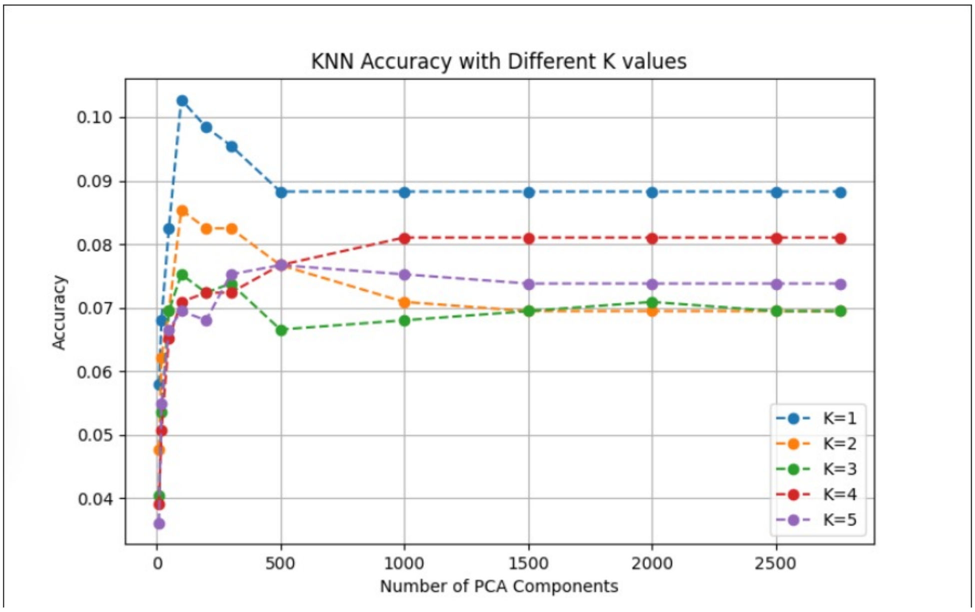


Figure 11: K-Nearest Neighbors Model Visualization with PCA

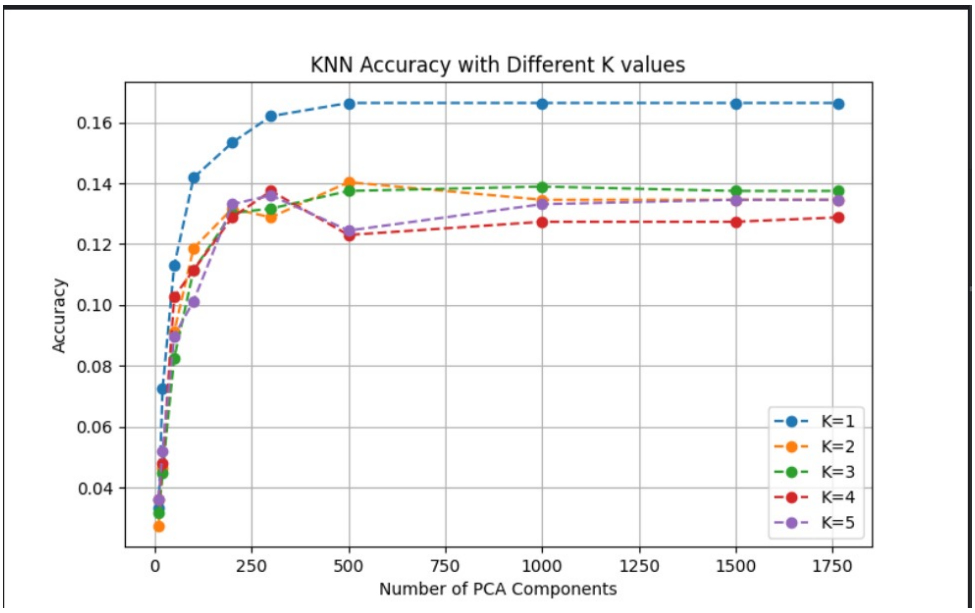


Figure 12: K-Nearest Neighbors Model Visualization with PCA+HOG

K-NN performed modestly, with K=1 consistently producing the best results across all feature configurations. This suggests that face classification benefits from very local decision boundaries, and increasing the neighborhood size dilutes the discriminative power. The combined PCA+HOG features significantly outperformed other configurations, achieving nearly double the accuracy of raw pixels alone.

4.5 AdaBoost

We implemented AdaBoost with multiple base estimators for comparison, including decision trees (max depth=1), SVM, and logistic regression.

4.5.1 AdaBoost with Logistic Regression

- Original pixels: 8.30% accuracy
- HOG features: 6.37% accuracy
- PCA: 9.84% (2500 components)
- PCA+HOG (500 components): 9.00% accuracy

4.5.2 AdaBoost with SVM

- Original pixels: 5.07% accuracy
- HOG features: 8.83% accuracy
- PCA: 5.50% (2000 components)
- PCA+HOG (100 components): 8.83% accuracy

4.5.3 AdaBoost with Decision Trees (max_depth=1)

- Original pixels: 2.46% accuracy
- HOG features: 2.89% accuracy
- PCA: 2.75% (20 components)
- PCA+HOG (300 components): 2.75% accuracy

AdaBoost ensemble models generally underperformed compared to their standalone counterparts, suggesting that the boosting approach may not be well-suited for this multi-class face classification task. The best-performing AdaBoost configuration used logistic regression with PCA+HOG features, but even this achieved only 9% accuracy. The poor performance may be attributed to AdaBoost's original design for binary classification rather than the multi-class problem (184 classes) we presented.

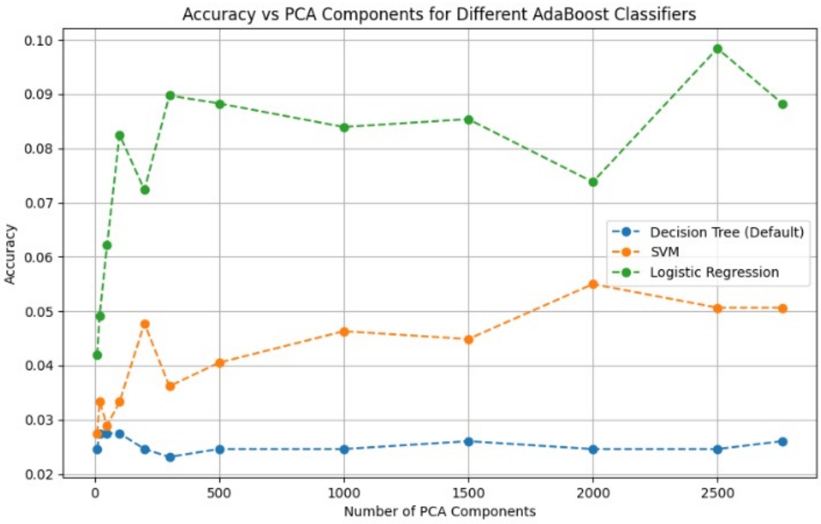


Figure 13: AdaBoost Model with PCA Visualization

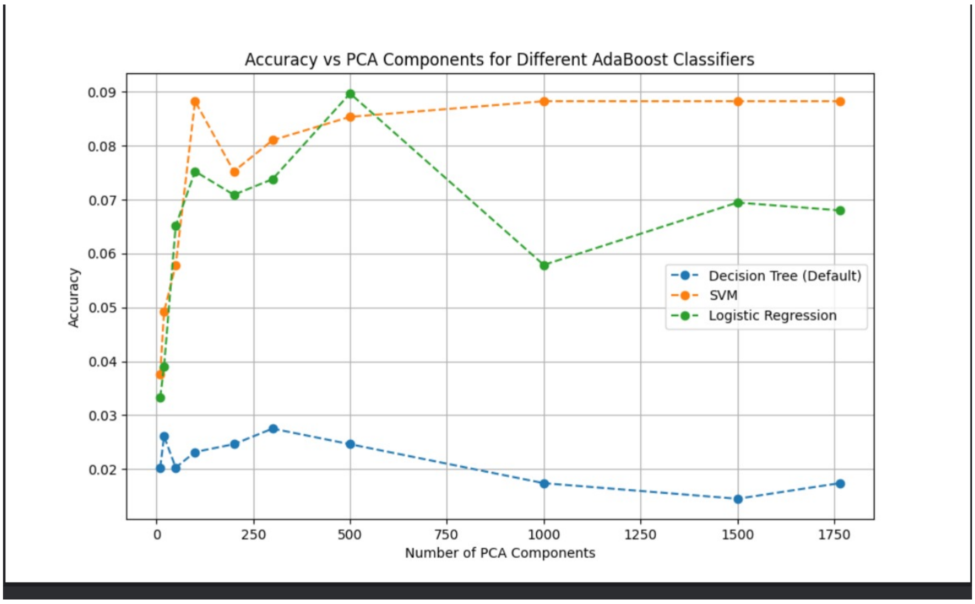


Figure 14: AdaBoost Model with PCA+HOG Visualization

4.6 Evaluation of Traditional Machine Learning Approaches

Our comprehensive evaluation of traditional machine learning approaches for face classification revealed several important patterns:

- **Feature Engineering Impact**: HOG features consistently improved performance across most models, demonstrating the importance of gradient-based features for capturing facial structures. The combination of PCA and HOG features generally yielded the best results, suggesting complementary information between global (PCA) and local (HOG) feature types.

- **Model Hierarchy**: Logistic regression emerged as the strongest traditional classifier (34% with PCA+HOG), followed by SVM (29.09%), K-NN (16.64%), and decision trees (9.70%). This hierarchy suggests that linear and margin-based models better capture the facial feature space than neighborhood or tree-based approaches.

- **Dimensionality Considerations**: Different models required different optimal PCA component counts, with logistic regression and K-NN performing best with 500 components, while SVM needed 1000-1500 components. This indicates varying abilities to leverage high-dimensional information.

- **Ensemble Limitations**: The poor performance of AdaBoost across all base estimators (best 9%) suggests that ensembling does not effectively address the complexities of multi-class face classification with limited data per class.

- **Performance Ceiling**: Even the best traditional model achieved only 34% accuracy, highlighting the inherent challenge of distinguishing between 184 classes with subtle differences and limited training examples.

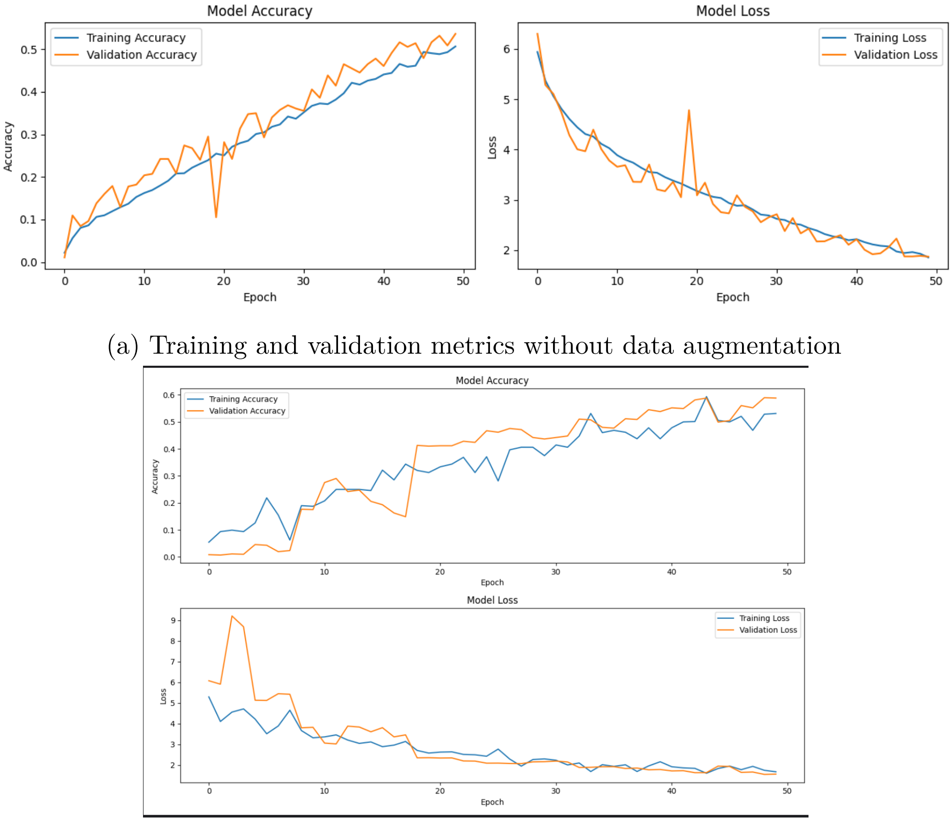
These results demonstrate that while traditional machine learning approaches can provide reasonable baselines, they fall significantly short of the performance achieved by deep learning models, particularly transfer learning approaches (66% with VGG16).

5 Deep Learning Approaches

5.1 Custom CNN Model

We designed a custom CNN architecture for face classification. Initially, without data augmentation, the model achieved a test accuracy of 53.62%. After implementing data augmentation, the final model improved to 58.97% test accuracy with the following configuration:

- Training samples: 2876
- Testing samples: 719
- Parameters:
  - Batch size: 32
  - Learning rate: 0.001
  - Data augmentation: rotation=10°, width/height shift=0.1, zoom=0.1, horizontalFlip=True



(a) Training and validation metrics without data augmentation

(b) Training and validation metrics with data augmentation

Figure 15: Comparison of model learning curves with and without data augmentation for custom CNN

5.1.1 Impact of Data Augmentation

Data augmentation significantly improved the performance of our custom CNN model, increasing accuracy from 53.62% to 58.97% (+5.35%). The augmentation strategy included:

- Rotation: ±10 degrees
- Width/height shift: 10% of total dimensions
- Zoom: 10% random zoom in/out
- Horizontal flip: enabled

Without augmentation, the model showed clear signs of overfitting. The introduction of augmentation techniques reduced this gap, though a noticeable difference between training and validation performance remained.

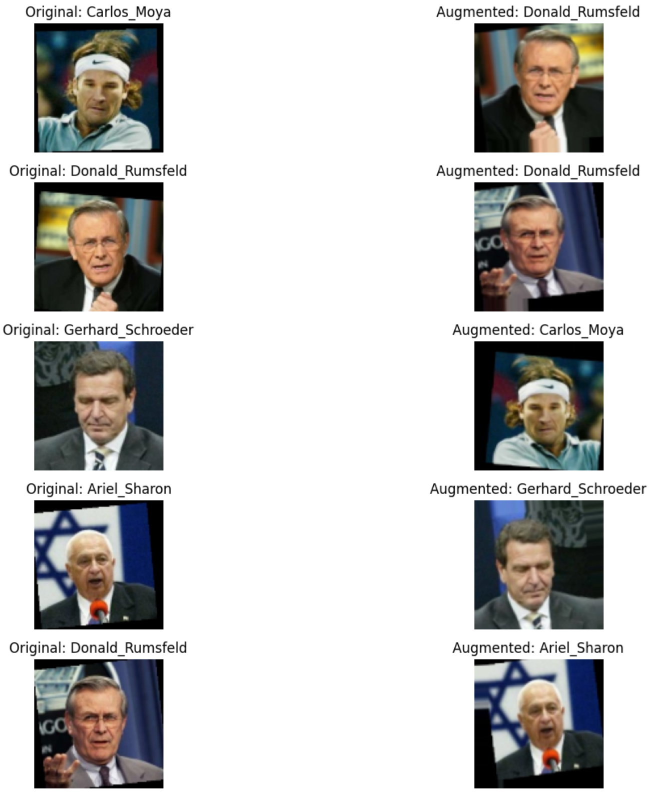

Original: Carlos_Moya
Augmented: Donald_Rumsfeld
Original: Donald_Rumsfeld
Augmented: Donald_Rumsfeld
Original: Gerhard_Schroeder
Augmented: Carlos_Moya
Original: Ariel_Sharon
Augmented: Gerhard_Schroeder
Original: Donald_Rumsfeld
Augmented: Ariel_Sharon

Figure 16: Examples of augmented face images using the specified transformation parameters

5.2 Transfer Learning with VGG16

Our primary deep learning approach utilized the VGG16 architecture with transfer learning:

5.2.1 Feature Extraction Phase

Initially, we froze the convolutional layers of VGG16 and trained only the classifier, achieving a validation accuracy of approximately 66%.

5.2.2 Fine-Tuning Phase

After initial training, we unfroze the last convolutional layers and fine-tuned the model with a lower learning rate, which improved performance significantly:

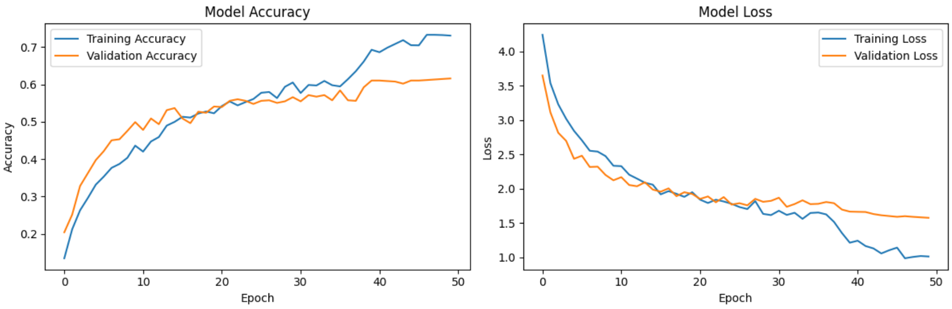


Figure 17: Training and validation metrics during the feature extraction phase of VGG16 transfer learning

- Training accuracy increased to approximately 78%
- Validation accuracy improved to around 66%
- Training loss decreased to approximately 0.8
- Validation loss stabilized at about 1.4

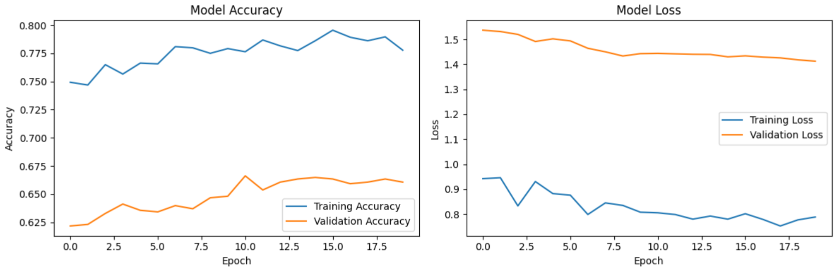


Figure 18: Training and validation metrics during the fine-tuning phase of VGG16 transfer learning

5.3 Key Deep Learning Techniques Employed

Our implementation leveraged several critical deep learning techniques to optimize model performance:

5.3.1 Activation Functions

- **ReLU (Rectified Linear Unit)**: Used throughout hidden layers to introduce non-linearity while avoiding the vanishing gradient problem common in traditional activation functions like sigmoid or tanh.
- **Softmax**: Applied in the output layer to convert raw logits into probability distributions across classes, essential for multi-class classification tasks.

5.3.2 Regularization Techniques

- **Dropout**: Implemented with rates of 0.25 after convolutional blocks and 0.5 in dense layers to prevent overfitting by randomly deactivating neurons during training, forcing the network to learn redundant representations.

- **Batch Normalization**: Applied after convolutional and dense layers to normalize activations, stabilize and accelerate training, and provide additional regularization benefits.
- **Data Augmentation**: Significantly improved generalization by creating novel training samples through controlled transformations.
- **Early Stopping**: Implemented with patience of 15 epochs to prevent overfitting by monitoring validation accuracy and stopping training when performance plateaued.

5.3.3 Optimization Strategies

- **Adam Optimizer**: Chosen for its adaptive learning rate capabilities, efficiently handling sparse gradients and noisy data compared to traditional SGD.
- **Learning Rate Management**:
  - Initial learning rate of 0.001 for custom CNN
  - Differential learning rates during VGG16 fine-tuning (1e-5 for convolutional layers, 1e-4 for classifier layers)
  - **ReduceLROnPlateau**: Dynamically reduced learning rates when performance plateaued, allowing finer optimization in later training stages
- **Categorical Cross-Entropy Loss**: Optimized for multi-class classification by measuring the difference between predicted probability distributions and one-hot encoded ground truth.

5.3.4 Transfer Learning Approach

- **Pre-trained Weights**: Leveraged VGG16 trained on ImageNet to transfer low-level feature detectors that generalize well across visual tasks.
- **Layer Freezing**: Initially froze convolutional base layers to preserve learned features while training only the classifier.
- **Selective Fine-Tuning**: Strategically unfroze deeper convolutional layers during fine-tuning to adapt high-level feature detectors to our specific face recognition task.
- **Custom Classifier**: Replaced the original VGG16 classifier with a task-specific architecture optimized for face classification.

5.3.5 Model Architecture Considerations

- **Increasing Filter Depth**: Progressively increased convolutional filters (32→64→128) to capture increasingly complex features while reducing spatial dimensions.
- **Multiple Dense Layers**: Implemented cascading dense layers (512→256→output) to learn hierarchical representations from extracted features.
- **Network Depth**: Balanced depth and computational efficiency with three convolutional blocks providing sufficient representational capacity while remaining trainable on available hardware.

These techniques worked synergistically to address the inherent challenges of face classification, with transfer learning and augmentation proving particularly effective given our dataset constraints. The performance gap between our custom CNN (58.97%) and fine-tuned VGG16 (approximately 66% validation accuracy) demonstrates the substantial benefits of leveraging pre-trained models for complex visual recognition tasks.

5.4 Analysis of Model Performance

5.4.1 Custom CNN Performance

The custom CNN model showed progressive improvement but faced inherent limitations:

- **Without augmentation**: The initial 53.62% test accuracy demonstrated the challenging nature of face classification with a custom architecture
- **With augmentation**: Improved performance to 58.97% (+5.35%) highlighted the effectiveness of data augmentation techniques
- Interestingly, in some training runs (Image 1), the validation accuracy slightly exceeded training accuracy in later epochs, suggesting good generalization despite the complexity of the task
- Both accuracy curves show a steady improvement throughout training, with final accuracy values around 52-59%
- Model loss decreased steadily from initial values above 5.0 to approximately 1.8-2.0

5.4.2 VGG16 Transfer Learning Performance

The VGG16-based model demonstrated superior performance through transfer learning:

- **Feature extraction phase**: Achieved approximately 66% validation accuracy by leveraging pre-trained convolutional features
- **Fine-tuning phase**: Training accuracy increased to approximately 78% while maintaining validation accuracy around 66%
- The gap between training and validation performance (78% vs. 66%) indicates some overfitting despite regularization
- The fine-tuned model showed better convergence with lower loss values (training loss of 0.8 vs. validation loss of 1.4)
- The superior performance of VGG16 over the custom CNN demonstrates the value of transfer learning for this task

5.5 Dataset Limitations

Despite the promising results, our dataset presented several notable limitations that likely impacted model performance:

- **Limited Diversity**: Many photos for the same individual exhibited similar attributes, often appearing to be from the same photoshoot or day, providing limited variation in pose, lighting, and expression.

- **Insufficient Data Volume**: While deep learning typically benefits from large datasets, our filtered dataset contained relatively few examples per person, constraining the model's ability to learn robust facial representations.
- **Temporal Inconsistency**: Some photos in the dataset were significantly outdated, with subjects' appearances having changed over time, introducing potential inconsistencies during training and evaluation.
- **Class Imbalance**: The number of photos varied significantly across individuals, potentially biasing the model toward better-represented classes.

These limitations highlight common challenges in face recognition tasks and the critical importance of high-quality, diverse training data for building robust recognition systems.

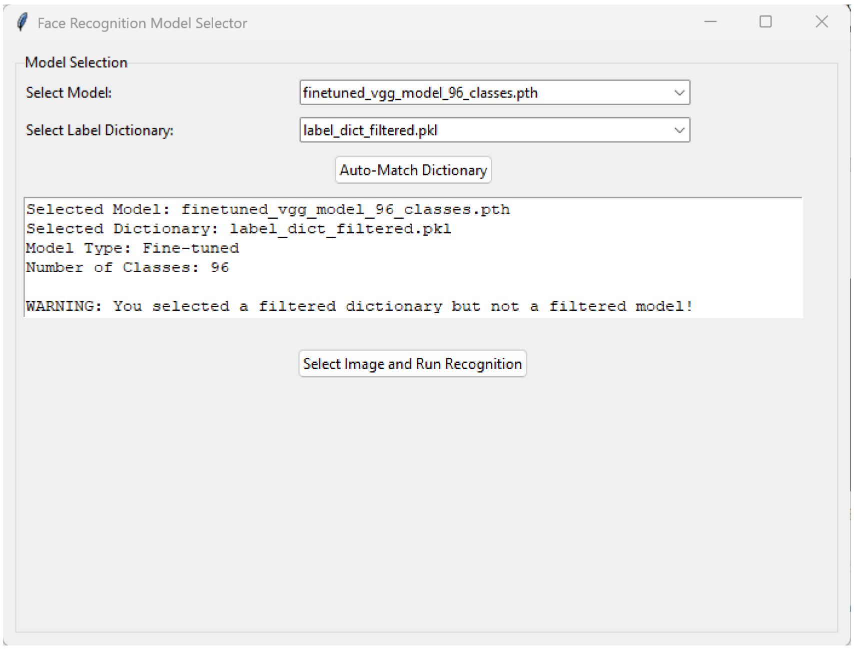
6 Face Recognition Application

6.1 Application Interface

To demonstrate the practical application of our trained models, we developed a user-friendly face recognition interface using Python's Tkinter library. The application, named "Model Selector Face Recognition," provides the following functionality:

- **Model Selection**: Users can choose between different trained models (custom CNN or VGG16 variants)
- **Label Dictionary Integration**: The interface automatically matches the appropriate label dictionary to the selected model
- **Face Detection**: Incorporates Haar Cascade classifiers from OpenCV to automatically detect and isolate faces in input images
- **Real-time Prediction**: Processes selected images and displays prediction results with confidence scores
- **Visual Feedback**: Shows both the original image with face detection overlay and the processed face region used for recognition

This application provides an intuitive way to test and compare different models on new images, enabling practical deployment of the trained face recognition systems. The modular design allows for easy integration of new models as they become available.



Face Recognition Model Selector
Model Selection
Auto-Match Dictionary
Select Image and Run Recognition

Select Model: finetuned_vgg_model_96_classes.pth

Select Label Dictionary: label_dict_filtered.pkl

```
Selected Model: finetuned_vgg_model_96_classes.pth
Selected Dictionary: label_dict_filtered.pkl
Model Type: Fine-tuned
Number of Classes: 96

WARNING: You selected a filtered dictionary but not a filtered model!
```

Figure 19: Screenshot of the Model Selector Face Recognition application interface

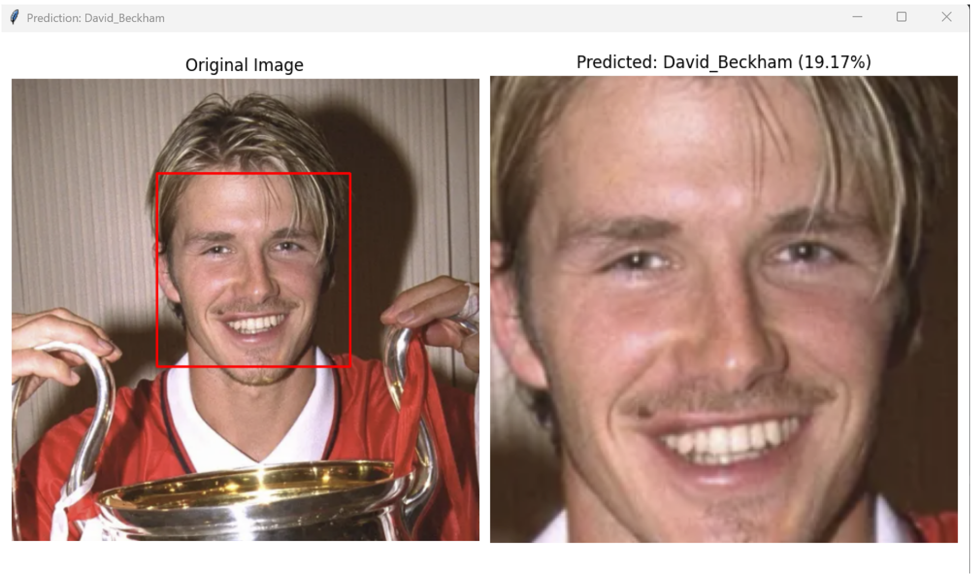


Figure 20: Trying the model on a photo of David Beckham from the internet

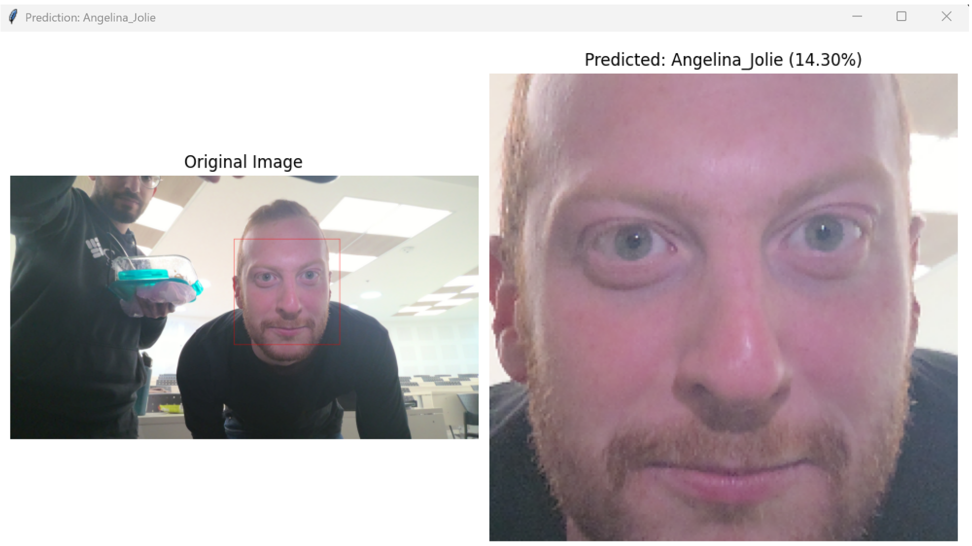

Prediction: Angelina_Jolie
Original Image
Predicted: Angelina_Jolie (14.30%)

Figure 21: Having fun with the model on our friends (They aren't part of the dataset - but could be in the future (:)

6.2 Error Analysis

We conducted a detailed analysis of misclassifications to understand the limitations of our models, particularly focusing on the CNN approaches. This analysis revealed several important patterns that provide insight into the challenges of face classification.

6.2.1 Temporal Factors in Misclassification

A significant finding from our error analysis was the impact of temporal variations on classification accuracy. We observed clear patterns related to the age of photographs:

- **Recent photographs**: The model struggled most with recent images of individuals, particularly when the training data predominantly contained older photographs. This suggests a temporal bias in the dataset toward historical appearances.
- **Historical consistency**: Images from similar time periods as the majority of training data were classified with higher accuracy, even when other challenging factors (lighting, pose) were present.
- **Age progression effects**: The model demonstrated difficulty recognizing the same individuals across significant age gaps, suggesting that facial aging introduces features that the model interprets as indicative of different identities.
- **Era-specific attributes**: Photographs from different decades often contain distinctive styling elements (hairstyles, makeup, photographic techniques) that may influence classification decisions beyond the actual facial features.

6.2.2 Additional Factors in Misclassifications

Beyond temporal variations, we identified several other common factors contributing to misclassifications:

- **Extreme pose variations**: Faces captured at unusual angles (particularly profiles and three-quarter views) were frequently misclassified
- **Occlusions**: Partial face coverage from sunglasses, masks, or other objects significantly reduced recognition accuracy
- **Lighting conditions**: Extreme lighting (very bright, very dark, or strong directional lighting) negatively impacted performance
- **Image quality**: Low resolution or heavily compressed images resulted in detail loss that affected classification
- **Similar-looking individuals**: The model occasionally confused people with similar facial features, particularly when they shared demographic characteristics
- **Limited training examples**: Individuals with fewer training images (more common in the Threshold-9 dataset) experienced higher misclassification rates

This analysis highlights the importance of temporal consistency in training data for face recognition systems intended to work across different time periods. It also suggests that applications requiring recognition of individuals across significant age ranges may benefit from specific age-invariant features or training with age-progressed examples.

7 Discussion

7.1 Model Performance Analysis

Our experiments yielded several notable findings:

- **Transfer Learning Superiority**: The fine-tuned VGG16 model significantly outperformed all other approaches, demonstrating the value of transfer learning from large pre-trained networks.
- **Traditional vs. Deep Learning**: While traditional methods achieved reasonable accuracy with HOG features, they fell short compared to deep learning approaches that automatically learn hierarchical features.
- **Dimensionality Reduction Trade-offs**: PCA improved computational efficiency but slightly reduced accuracy, suggesting a trade-off between performance and resource usage.
- **Feature Extraction vs. Fine-Tuning**: The performance boost from fine-tuning (3.1% increase) confirms the value of adapting pre-trained convolutional filters to the specific dataset.
- **Dataset Threshold Impact**: The number of photos per person critically affected model performance, with both CNN models performing better on the Threshold-15 dataset despite having fewer total classes.

7.2 Impact of Minimum Photo Threshold

The comparison between Threshold-9 and Threshold-15 datasets revealed important insights:

- **Quality vs. Quantity Trade-off**: Having more photos per person (Threshold-15) led to better model performance than having more individuals with fewer photos each (Threshold-9). However, maybe more quality and unique photos of the person would outperform over quantity.
- **Model Sensitivity**: Traditional models showed greater sensitivity to the reduction in per-person photos compared to deep learning approaches.
- **Generalization Capability**: Models trained on the Threshold-15 dataset exhibited better generalization to test images, suggesting that a richer representation per individual is more valuable than having a wider variety of individuals.
- **Training Efficiency**: The Threshold-15 dataset allowed for faster training and convergence despite the reduced class diversity.

These findings highlight the importance of data quality and representation balance in face recognition tasks, suggesting that practitioners should prioritize collecting multiple high-quality images per individual rather than expanding the number of individuals with limited samples.

8 Conclusions

This study compared multiple machine learning approaches for face classification using a modified LFW dataset. Our comprehensive evaluation of traditional and deep learning models yielded several key findings:

- Transfer learning with VGG16 significantly outperformed all other approaches, demonstrating the effectiveness of leveraging pre-trained networks for face classification tasks.
- Feature extraction techniques, particularly the combination of HOG and PCA, substantially improved the performance of traditional machine learning models.
- Data quality proved more important than quantity, with models trained on fewer classes but more examples per class (Threshold-15) consistently outperforming those trained on more classes with fewer examples each (Threshold-9).
- Data augmentation provided significant performance improvements for deep learning models, helping to address the limited training examples per class.
- Temporal consistency in training data emerged as a critical factor, with models struggling to recognize individuals across significant age differences.

The substantial performance gap between traditional machine learning approaches (best: 34% with Logistic Regression) and deep learning models (best: 66% with VGG16) underscores the power of deep neural networks for complex visual recognition tasks. However, our error analysis revealed persistent challenges in handling temporal variations, extreme poses, and occlusions, suggesting areas for future improvement.

For practical applications, our findings suggest that practitioners should prioritize: (1) collecting multiple diverse images per individual, (2) utilizing pre-trained models with transfer learning when possible, and (3) implementing robust data augmentation to artificially increase training sample diversity. Future work could explore more sophisticated approaches to address temporal variations in facial appearance, potentially through age-invariant feature extraction or targeted data augmentation strategies.

Model	Original Pixels	HOG	PCA	PCA+HOG
Logistic Regression	28.80%	31.84%	27.00%	**34.00%**
SVM	21.85%	28.80%	22.00%	29.09%
Decision Tree	**9.70%**	4.92%	3.90%	3.04%
K-NN (K=1)	8.80%	10.67%	10.27%	16.64%
AdaBoost (LogReg)	8.30%	6.37%	**9.84%**	9.00%
Custom CNN	58.97% (with data augmentation)			
VGG16 Transfer Learning	**66.00%** (fine-tuned)			

Table 2: Performance comparison of all implemented models. Best performance for each model type is highlighted in bold.

8.1 Future Work

Based on our findings, future research could explore:

- Experiment with another advanced CNN architectures like ResNet, EfficientNet.
- One-shot or few-shot learning approaches for individuals with limited training samples, particularly relevant for the many individuals in LFW with only one or two photos
- Model compression techniques to make deep learning models more deployable on resource-constrained devices
- Cross-dataset evaluation to assess how well models trained on LFW generalize to other face datasets
- Enlarging the dataset with more unique photos per individual to enable more precise classification, including potentially adding photos of friends and family for personalized recognition
- Expanding to a larger dataset with more classes to develop a mobile application that can determine which celebrity a user most resembles
- Developing an unsupervised face clustering model without predefined classes, similar to modern smartphone photo organization systems, where faces without assigned names can still be meaningfully grouped