Facial Recognition AI Pet Dog

CMSE890 AML Final Project

Yuhan Zhu

2024/04/17

The “Business” Objective

Develop a responsive synthetic pet dog that enhances security by distinguishing between familiar and unfamiliar faces and reacting appropriately to unknown faces.

The goal can be break down into these objectives:

- **Security**: Provide a basic level of security by alerting the owner to the presence of someone unfamiliar.
- **Simplicity in Interaction**: Keep interactions straightforward, such as making a distinct sound when an unknown face is detected.
- **Adaptability**: adapt to new faces as they become familiar, without requiring manual updates to its database of known individuals.

Initial Thought

- Train ML models like CNN, SVM, KNN
- Tuning hyper parameters
- Use open source facial recognition packages
- Performance metrics: AUC ROC, TPR, and TNR
- Cross-validation to evaluate the model performance

Overall Workflow

- Data IDA,
- Data EDA,
- Preprocessing Data,
- Data Scaling, PCA
- Training the Models,
- Evaluating the models,
- tuning the models

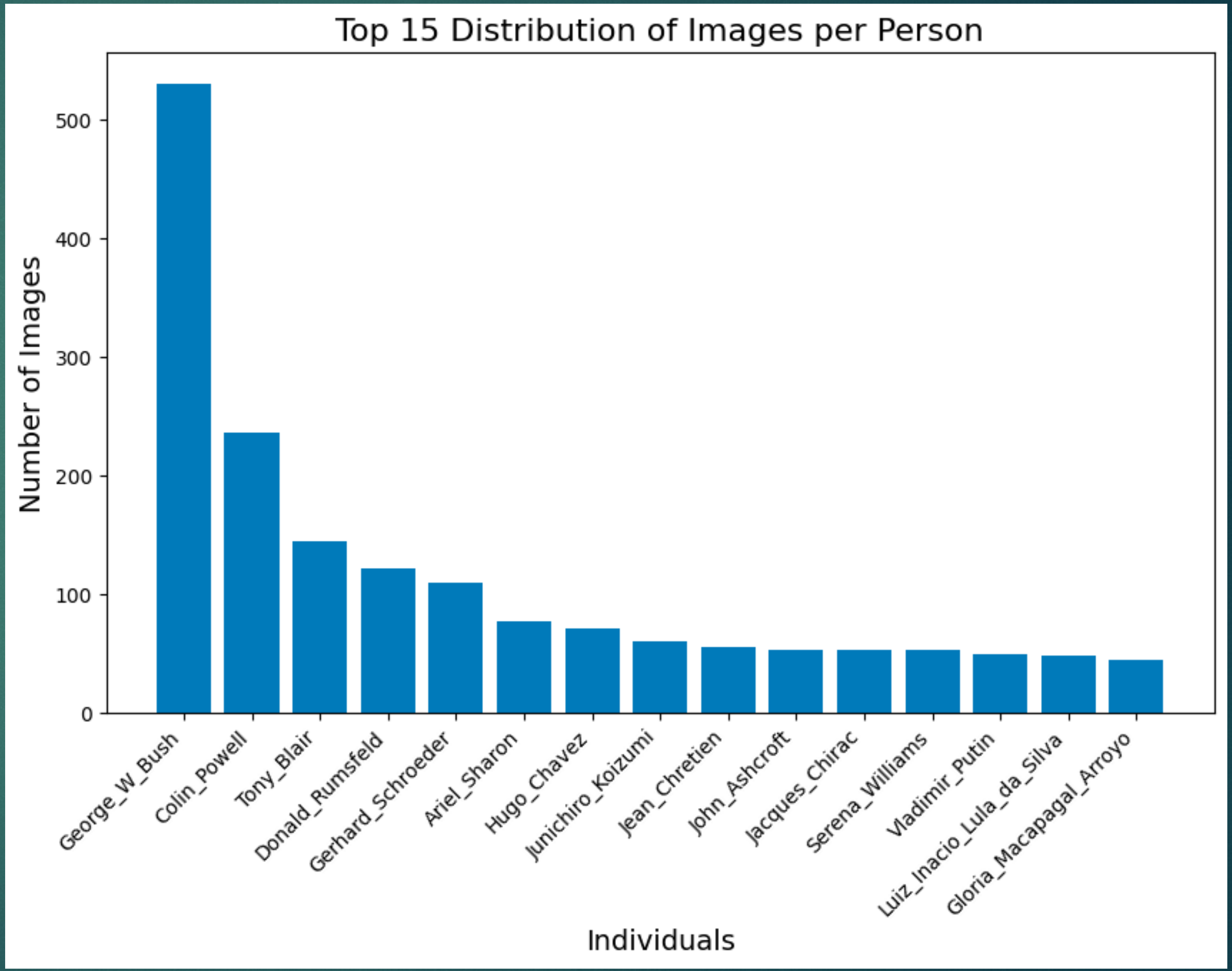
Data Selection

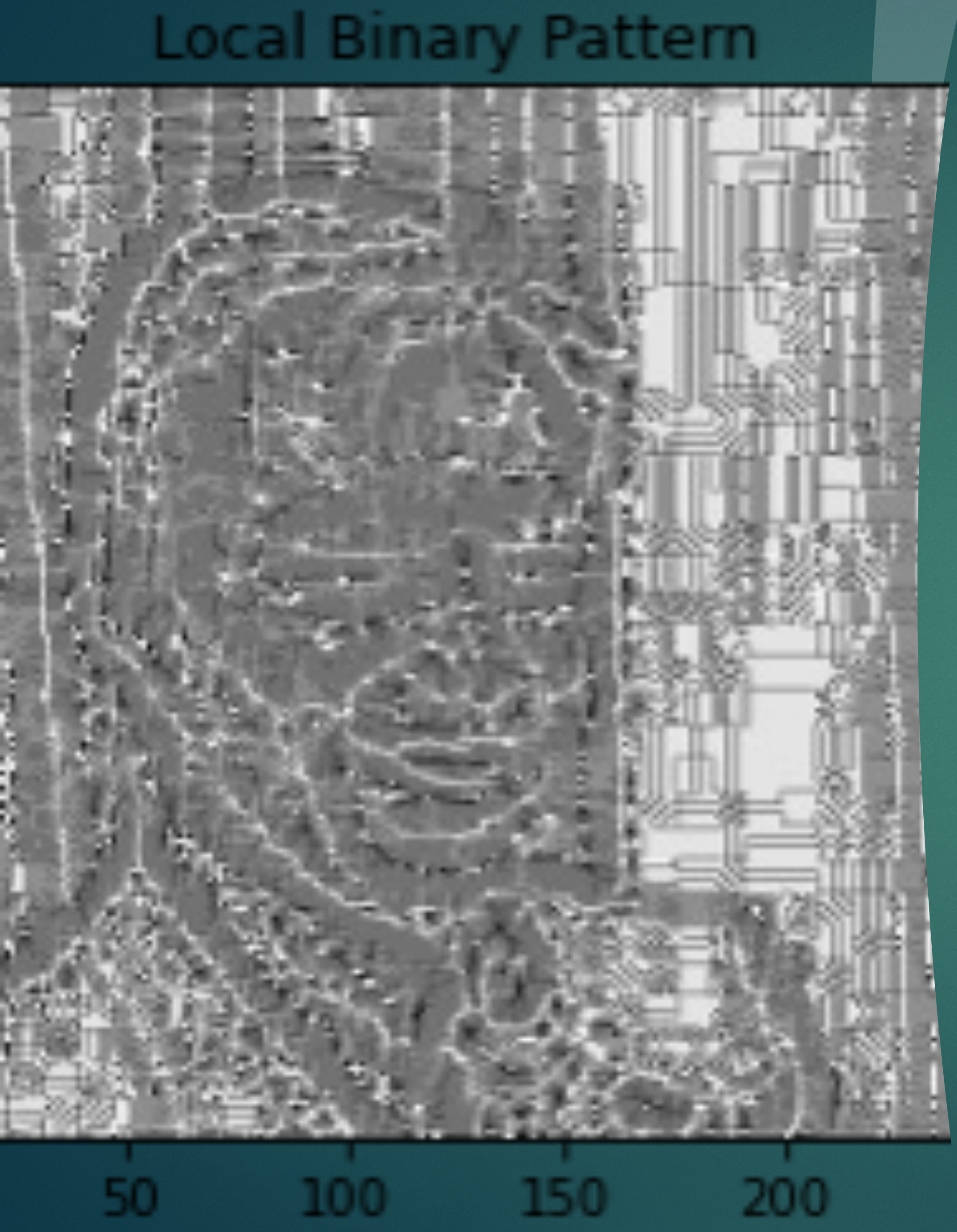
Labeled Faces in the Wild: All images aligned with deep funneling

Total number of images: 13233 Number of classes (individuals): 5749

Nature of the LFW Dataset

The dataset is naturally imbalanced; some individuals have many pictures, while others have very few.



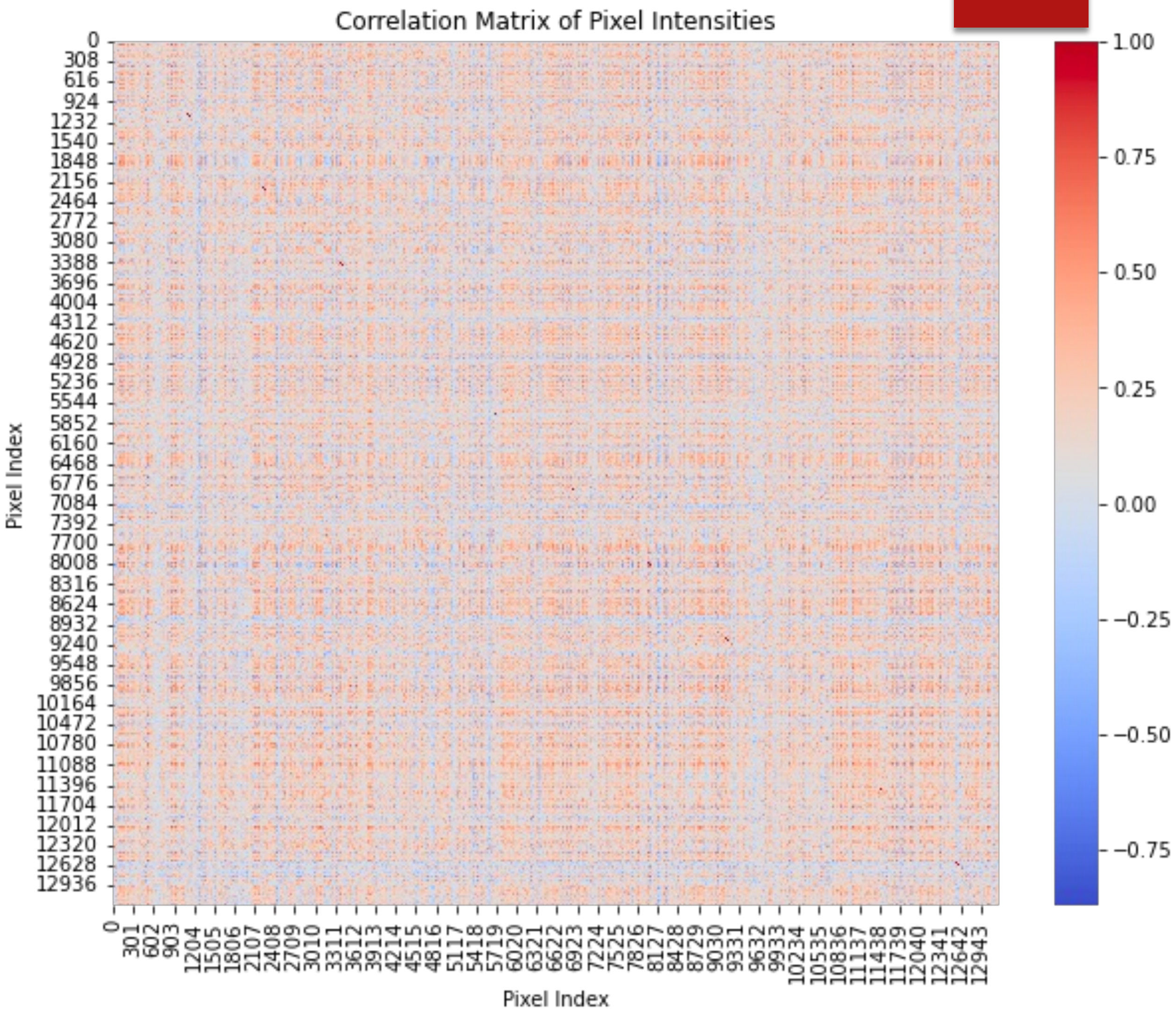


Identifying features under various conditions.

- The photos have variations in lighting, facial expressions, and poses.
- Faces are not perfectly aligned in the LFW dataset.
- Some faces may be occluded by glasses, hats, or other objects.

Correlation Matrix of Pixel

Identify if there are any high correlations between different pixel regions or features within the images that might suggest redundancy.



Pre-processing Data Workflow

- Reduces Image Complexity: Converts images to grayscale.
- Resizes images to 250x250 pixels.
- Transforms 2D image matrix into a 1D array, suitable for SVM and KNN
- Normalization
- Label Encoding
- train-validation-test splits

Convolutional Neural Network (CNN) Model

Baseline Model

- Conv1: 32 feature, 3*3 kernel size
- Conv2: 64 feature, 3*3 kernel size
- Pooling1: 2*2
- Dropout1: 0.2
- FullConnect
- Dropout2: 0.5

```
score = base_model.evaluate(X_test, y_test, verbose=0)
print('Baseline Test accuracy: {0:%}'.format(score[1]))
```

```
Baseline Test accuracy: 70.300001%
```

Convolutional Neural Network (CNN) Model

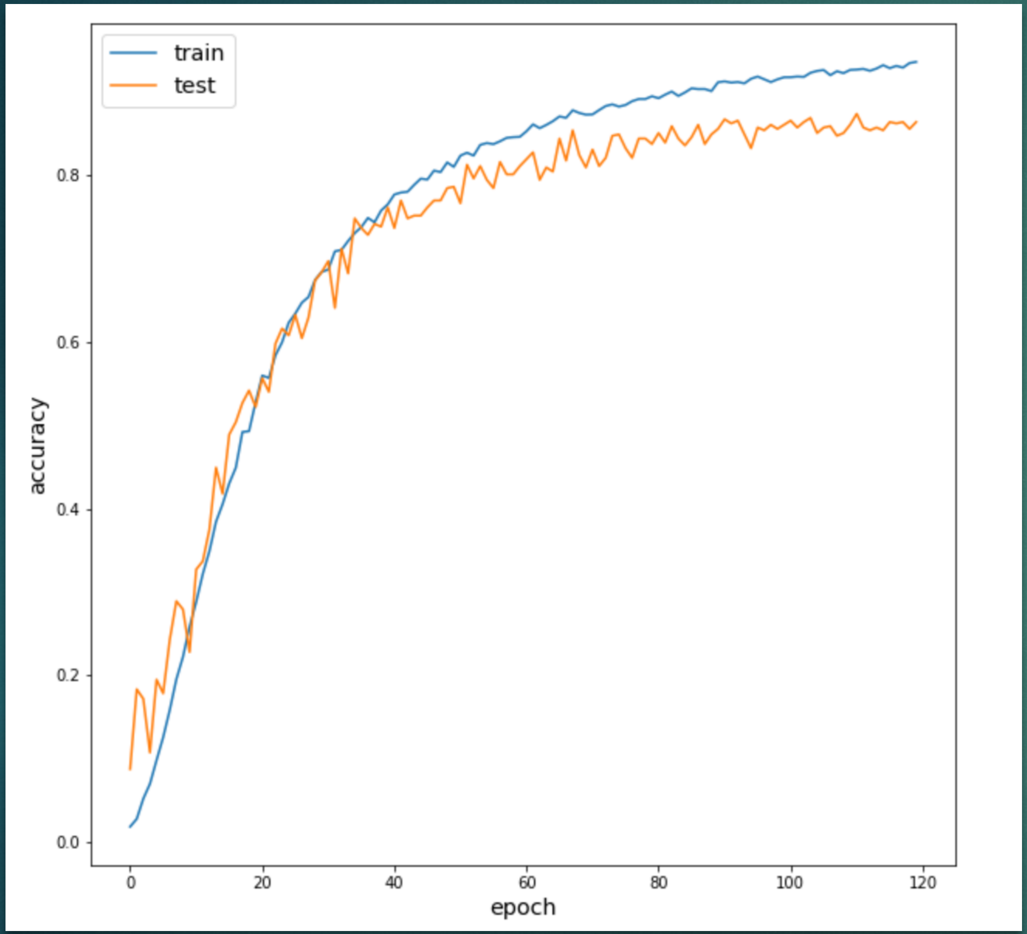
Model A

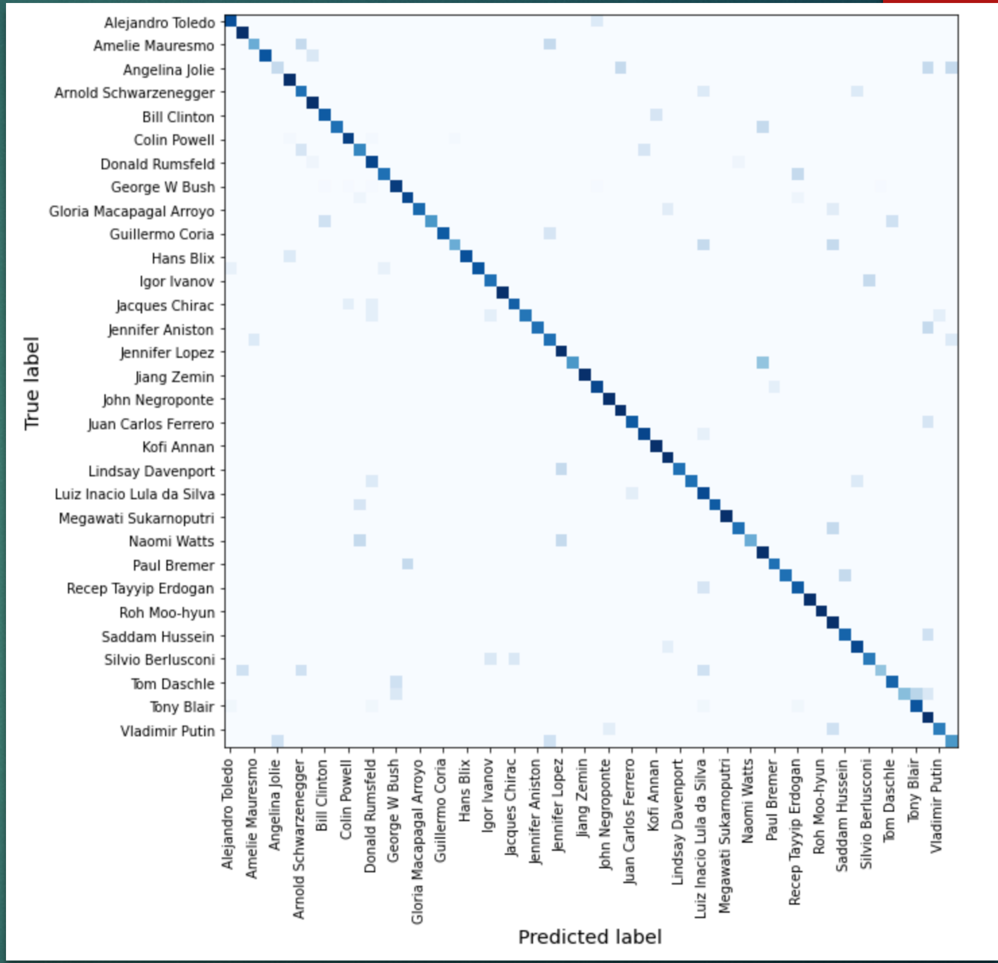
- Conv1: 32 feature, 5*5 kernel size
- Conv2: 64 feature, 5*5 kernel size
- Pooling1: 2*2
- Dropout1: 0.2
- FullConnect
- Dropout2: 0.5

```
score = modelA.evaluate(X_test, y_test, verbose=0)
print('modelA Test accuracy: {0:%}'.format(score[1]))
```

```
modelA Test accuracy: 69.999999%
```

Convolutional Neural Network (CNN) Model





Batch_size = 256 test_split = 0.2 base_lr = 0.001 epochs = 120 ~85% accurcy
np.Mean(recall)=0.8141190958270488 np.mean(precision)=0.8486846927215296

Support Vector Machines (SVM)

```
Best estimator found by grid search:
SVC(C=1000.0, break_ties=False, cache_size=200, class_weight='balanced',
    coef0=0.0, decision_function_shape='ovr', degree=3, gamma=0.005,
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
Predicting people's names on the test set
done in  0.076s
                   precision    recall  f1-score   support

     Ariel Sharon       0.86      0.46      0.60        13
     Colin Powell       0.80      0.87      0.83        60
  Donald Rumsfeld       0.94      0.63      0.76        27
    George W Bush       0.82      0.98      0.89       146
Gerhard Schroeder       0.95      0.80      0.87        25
      Hugo Chavez       1.00      0.53      0.70        15
       Tony Blair       0.97      0.78      0.86        36

         accuracy                           0.85       322
        macro avg       0.91      0.72      0.79       322
     weighted avg       0.86      0.85      0.84       322

Confusion Matrix is:
[[  6   2   0   5   0   0   0]
 [  1  52   0   7   0   0   0]
 [  0   2  17   8   0   0   0]
 [  0   3   0 143   0   0   0]
 [  0   1   0   3  20   0   1]
 [  0   4   0   2   1   8   0]
 [  0   1   1   6   0   0  28]]
```

Model comparison

- CNN can automatically and effectively learn spatial hierarchies of features (such as edges and textures of faces) through their deep and structured network layers, which are specifically designed to capture and exploit the structural and compositional nature of images.
- SVM are effective in high-dimensional spaces and are capable of complex decision boundaries with a relatively simple mathematical formulation, using kernel tricks to handle non-linear relationships efficiently.
- KNN is simple, intuitive, and efficient.
- All the model can reach 80+% accuracy

After Thoughts: Using the face_recognition Package

This package can be used to quickly build a prototype. It is based on C++ and dlib.

"The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark." https://github.com/ageitgey/face_recognition/blob/master/README.md

- **Workflow:**

1. Preprocess the LFW dataset to detect faces and extract encodings using face_recognition.
2. Store face encodings and corresponding labels (person's name) in a database.
3. When a new image is encountered, detect the face and extract its encoding.
4. Compare this encoding with those in the database using face_recognition.compare_faces to determine familiarity.
5. If no match is found, the system signaling as an unfamiliar face.

Conclusion

- Using CNN, SVM, and KNN are great approach for these types of problems, the machine learning solution is valuable.

- There are well-designed, open-source packages available that can perform same job. However, understanding the fundamentals is important.

- With the trained model, we can achieve approximately 80% accuracy in recognizing faces.

Discussion

- Future enhancements could include real-time adaptive learning mechanisms to refine recognition capabilities continuously.

- Adding video recognition capabilities.

- Incorporating a camera, speaker, and motor.

Thank you for listening!