Result and Summary

```
Number of augmented training images: 256
Number of augmented training labels: 256
```

```
[ ] print(f"Number of testing images: {len(X_test)}")
    print("Displaying testing images:")
```

```
Number of testing images: 32
Displaying testing images:
```

Summary

In this project, we aimed to develop an image classification model using a Convolutional Neural Network (CNN) to differentiate between images of two categories: your own images and those of a friend. Here's a summary of the steps involved:

1. **Dataset Preparation:**
    - We began by loading and preprocessing images from a dataset. Images were resized to a consistent dimension (128x128 pixels) and labeled according to their source (either your images or your friend's images).
    - The dataset was augmented to increase its size and variability. Augmentation techniques included rotation, shifting, shearing, zooming, and flipping.
2. **Data Augmentation:**
    - To enhance the robustness of the model, we used data augmentation techniques to generate a larger training dataset. The augmentation process produced 256 training images from an initial set of 20 images.
3. **Model Building:**
    - A Convolutional Neural Network (CNN) was constructed with multiple convolutional layers, max-pooling layers, and dense layers. The final output layer used a softmax activation function to classify images into two categories.
    - The model was compiled with the Adam optimizer and categorical cross-entropy loss function.
4. **Training and Evaluation:**
    - The model was trained on the augmented dataset with a batch size of 16 and for 20 epochs. The performance of the model was evaluated using a separate testing set of 32 images.
    - Training history was monitored to assess the model's performance over epochs.
5. **Results and Visualization:**
    - The number of augmented training images and testing images was displayed. A subset of these images was visualized to ensure the dataset's integrity.
    - The final trained model was saved for future use.

Conclusion

This project successfully demonstrated the process of building and training a CNN for image classification using a relatively small dataset that was augmented to achieve the desired number of training images. Key takeaways include:

1. **Data Augmentation:** Augmenting the dataset significantly increased the number of training examples, which is crucial for training effective deep learning models when working with limited data.
2. **Model Effectiveness:** The CNN model was able to learn features from the augmented images and classify them into the two categories with a reasonable accuracy. The final model was evaluated on a separate test set, providing an indication of its generalization capability.

3. **Future Work:** While the current model performs well, there is room for further improvement. Additional techniques such as fine-tuning, hyperparameter optimization, and the use of more complex architectures could enhance model performance. Moreover, expanding the dataset with more varied images and labels could improve the model's robustness and accuracy.

Overall, this project highlights the importance of data preparation and augmentation in training deep learning models and provides a foundation for further exploration and refinement in image classification tasks.