Family Facial Recognition for Home Security Applications

COE 379L Final Project Report

Software Design for Responsible Intelligent Systems

Nikhil Sharma and Constantinos Skevofilax

May 3, 2024

Introduction

Residential neighborhoods often face security risks such as burglaries, vandalism, and trespassing. These issues not only result in property loss and damage but also erode the sense of safety among community members. Traditional security measures like neighborhood watches and basic alarm systems are helpful, but they sometimes fall short in both deterrence and in providing clear, actionable evidence after a crime has occurred. The limitations of these methods, such as the need for constant human presence and often grainy video quality, make them less effective at identifying perpetrators and preventing crime.

Advancements in home security technology, particularly Ring cameras and facial recognition software, offer a more robust solution to these problems. Ring cameras provide high-resolution video surveillance that can be accessed remotely from smartphones, allowing homeowners to monitor their home from any location.

When combined with a machine learning algorithm trained to detect known faces, these systems can alert homeowners and law enforcement to the presence of known offenders or anyone who is not recognized as a known face, enhancing preventive measures. The integration of machine learning in a Ring camera not only helps in catching anyone more efficiently but also acts as a strong deterrent, as potential thieves are aware of the increased risk of identification and capture if they are not recognized by a system.

Project Statement

The goal of this project is to create a machine learning model that can help us identify a familiar face at a residence or a stranger who may be a source of suspicion. The model will be trained on pictures of Costaki's family members who live at his house as well as a dataset of strangers who do not live at his house. Our goal is for our models to accurately and efficiently tell us who the specific family member is or if it is a stranger.

We will be creating two machine learning models—a convolutional neural network (CNN) and a convolutional neural network with LeNet-5 architecture (LeNet-5)—to test our data on them. We chose a standard convolutional neural network to show the simplest form of neural network architecture and its limitations, and chose a more refined LeNet-5 model to improve our result accuracy as it is an extremely strong model in image recognition.

Data Sources

For the images of the “known” set, the data consists of photographs of members of Costaki’s immediate family—himself, his sister, his mother, and his father. Each photograph is a headshot of a family member looking at the camera to maximize the model’s ability to detect unique features and differentiate between family members who have similar facial features. Each family member had around 50 pictures each, making the known data set around 200 images. To increase our sample size, we decided to augment our dataset, which we go into detail about in the next section.

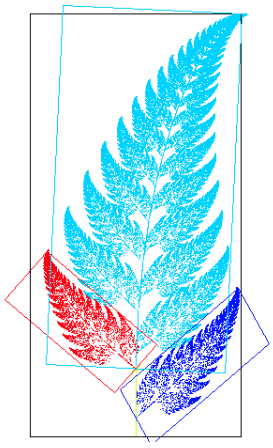
The data source for the “unknown” set of people, or the strangers, is a dataset from the Internet titled Labeled Faces in the Wild and contains 1555 images of famous celebrities but are not part of Costaki’s family. Each image has a standard size of 250 by 250 pixels. The purpose of adding these “unknown” faces is to help the model see faces that are not supposed to be detected by a camera.

Data Augmentation

Because the size of the strangers dataset was significantly larger than the family data set (~50 images per person), we decided to augment our family set to create a large enough sample size that would match the size of the stranger set.

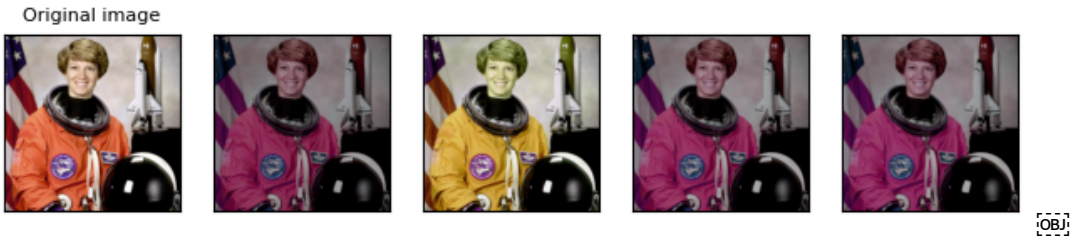
For augmentation, we decided to use TensorFlow’s built in `transforms()` function and opted for the following transformations:

1. **Affine Transformation:** translates the image horizontally randomly at a uniform range of [-0.5, 0.5) multiplied the width of the image.



Affine Transformation

2. **Random Color Jitter:** randomly adjusts the brightness of the image within a range of [0, 1] and converts the image to PyTorch tensor for processing.



Random Color Jitter

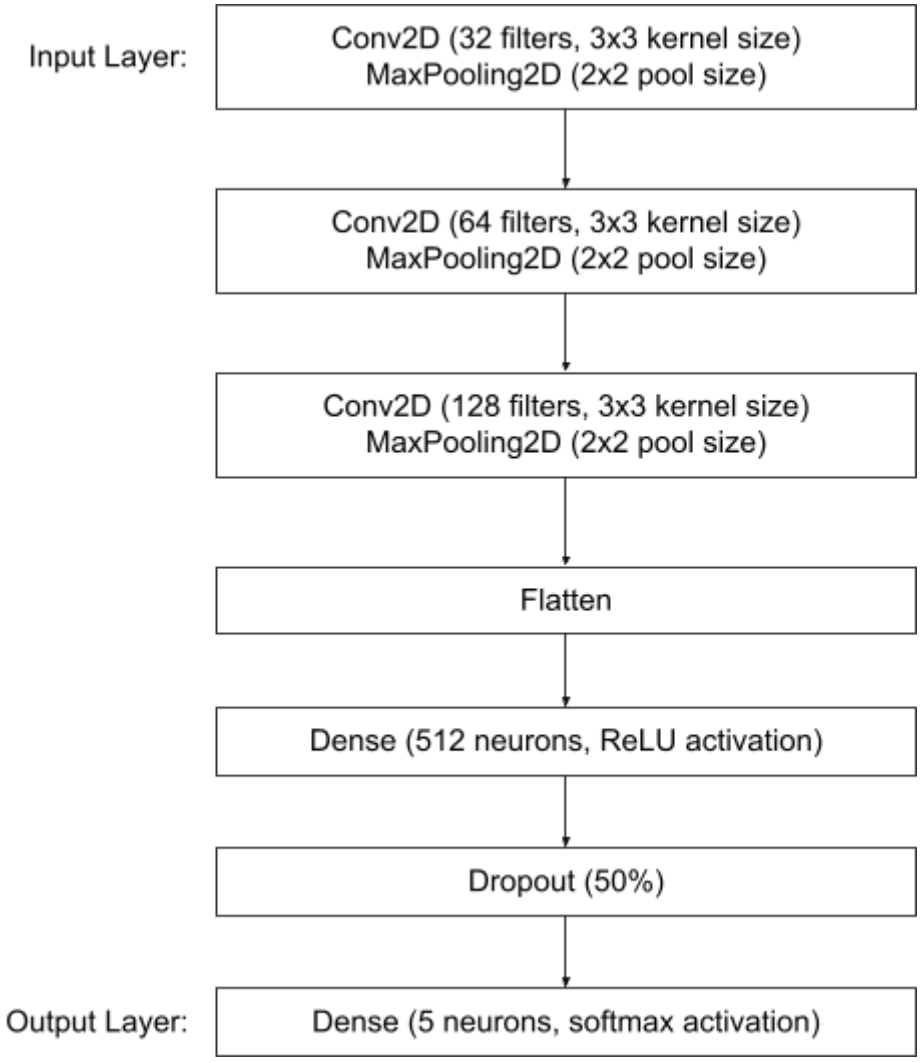
3. **Random Affine Transformation with Rotation:** We also opted to use the affine transformation with rotation on a range of [-30, 30) degrees to create more augmentation in how the images would potentially appear to the model via the camera.

Sample Affine Transformation with Rotation from Family Dataset

Models

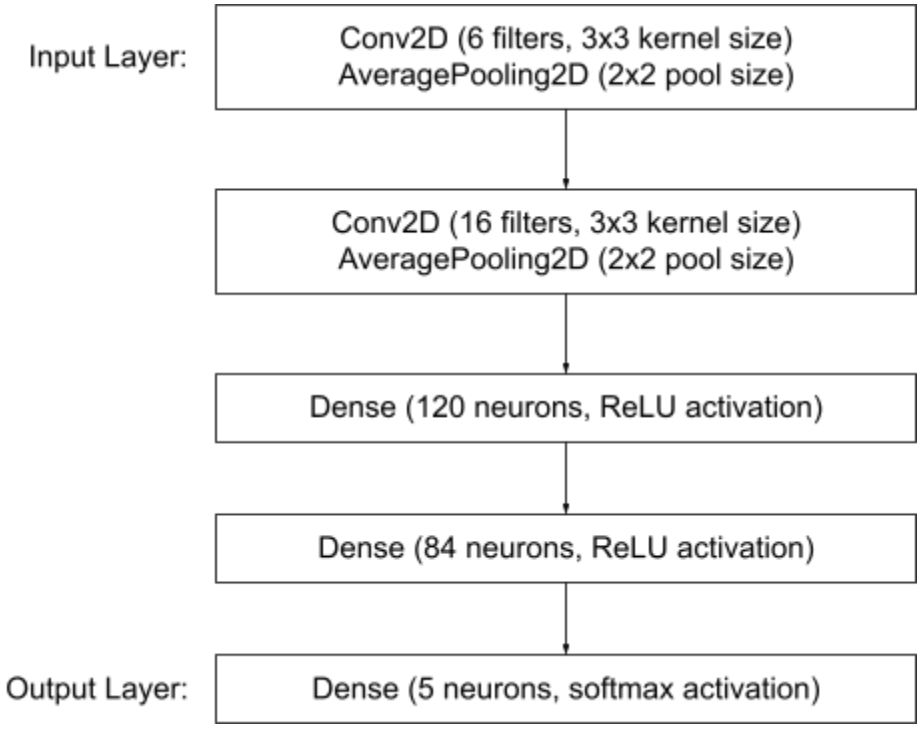
Standard Convolutional Neural Network

In convolution neural networks, the layers are typically designed so that the depth of feature maps increases deeper into the network. This structure allows the network to capture increasingly complex and abstract features of the input data. In the case of image data, lower layers capture basic features like edges and simple textures, middle layers might capture more complex features like parts of objects, and higher layers can capture high-level features that represent entire objects, such as facial features. For this reason, the filters in the 2-dimensional convolutions are 32, 64, and 128 for the first 3 layers. Once the layer is flattened, another fully connected layer is applied with 512 neurons and ReLU activation. A dropout layer is applied with a rate of 0.5, and a final dense layer of size 5 is added to be the output layer with softmax activation.



Convolutional Neural Network with LeNet-5 Architecture

For the LeNet-5 model architecture, we decided to use 5 layers—an input layer, 3 intermediate layers, and an output layer of size 5. The first two layers saw 2-dimensional convolutions with a kernel size of 3x3 and an average pooling size of 2x2. We chose 6 and 16 filters for these layers, respectively, because we thought it would be fair for the amount of different people the model would see. Three dense layers were added at the end with sizes of 120, 84, and 5. The first two dense layers had ReLU activation while the final output layer had softmax activation. Because facial features and characteristics vary between people, with additional transformations and filters on top due to augmentation, there would be more than enough variance to get more accurate test results.



Results

To distinguish between the family members and the strangers, we used the following class indices, which are referenced throughout the paper as follows:

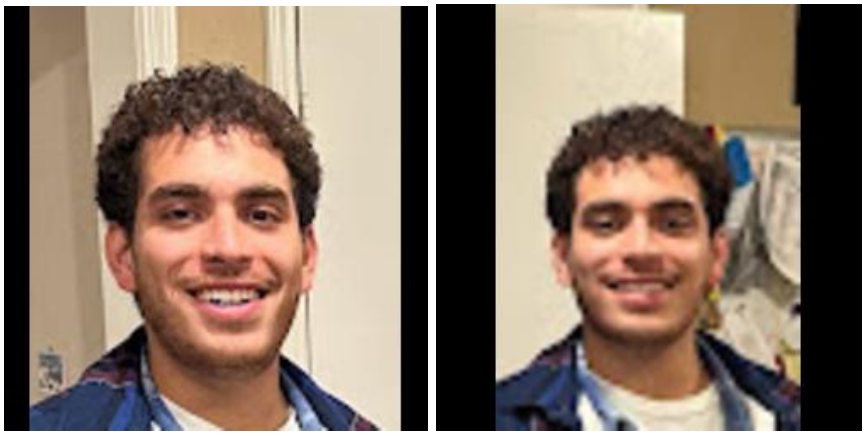
```
[[athena, costaki, george, stranger, teresa]]
```

Regular Convolutional Neural Network

An observation that we made is that the quality of our images heavily impacted the model's prediction ability. A clear picture with sunlight and high resolution is able to predict the individual with a very high accuracy, but a grainier, darker with no contrast picture is more susceptible to confusion between family members or labeling a known family member as a stranger. The model appears to work with more accuracy on images that are not rotated.

Below are three examples of pictures with high accuracy on predictions and one example of a picture with low accuracy.

Strong examples:



REGULAR-CNN (92.82% Sure is Costaki - CORRECT):
[[9.6351681e-03 9.2824227e-01 9.1461441e-04 3.2767978e-02 2.8439982e-02]]

REGULAR-CNN (99.8% Sure is Costaki - CORRECT):
[[4.893578e-07 9.989728e-01 7.280813e-05 9.540226e-04 5.639842e-09]]

Costaki Test Sample

REGULAR-CNN (99.5% Sure is Costaki - CORRECT):
[[1.3416535e-05 9.9552971e-01 2.5327338e-04 6.1089249e-09 4.2036278e-03]]
LENET-5 (78.32% Sure is Costaki - CORRECT):
[[8.6723996e-04 7.8316939e-01 2.1423160e-01 9.6195744e-04 7.6976698e-04]]

Weak example:

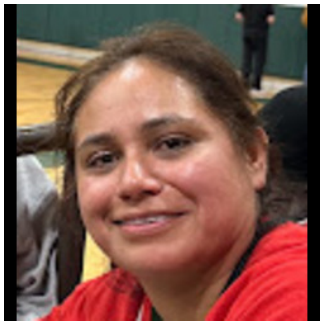
REGULAR-CNN (0.011748859% Sure is Costaki - INCORRECT):

[[1.4535964e-01 1.1748859e-04 2.5134797e-03 4.1054976e-01 4.4145963e-01]]

We see in the first two pictures, the model is 92.82% and 99.90% confident that the person in the images is Costaki, respectively. In the weak example, the model has almost 0 confidence that the person in the image is Costaki and thinks he is a stranger (41.05%) or his mother (44.15%).

Convolutional Neural Network with LeNet-5 Architecture

The LeNet-5 model yielded extremely accurate results for some instances, despite a test accuracy value of roughly ~0.2233. When individual images were tested, the model often yielded results with ~99% confidence. This model was able to handle a wider range of pictures, no matter the lighting, grain, or distance between the person and the image frame compared to the regular CNN.



Teresa Test Sample

LENET-5 (99.8% Sure is Teresa - CORRECT):

[[5.3695892e-04 9.6930613e-08 4.1745522e-04 8.2661060e-04 9.9821883e-01]]

REGULAR-CNN (65.23% Sure is Teresa - CORRECT):

[[3.0373386e-03 4.0310288e-06 3.4223872e-01 2.4456459e-03 6.5227425e-01]]

George Test Sample

LENET-5 (86.37% Sure is George - CORRECT):

[[1.1473622e-05 9.2283524e-05 8.6356109e-01 1.3626181e-01 7.3441995e-05]]

REGULAR-CNN (17.32% Sure is George - INCORRECT GUESS):

[[2.9306380e-05 1.7576041e-03 1.7323026e-01 5.8851402e-02 7.6613140e-01]]

Athena Test Sample

LENET-5 (99.9% Sure is Athena - CORRECT):

[[9.99081373e-01 1.64522729e-09 9.16093588e-04 2.33646620e-06 1.15430566e-07]]

REGULAR-CNN (80.56% Sure is Athena - CORRECT):

[[0.80565846 0.00406413 0.06941795 0.09734685 0.02351256]]

However, when running other tests, it appears that the LeNet-5 model will most likely classify a family member as a stranger when it cannot categorize a family member correctly. In comparison, the regular CNN model will tend to find facial features that match a family member, such as matching Athena (the daughter of Teresa) with Teresa, and confuse the two. This makes more sense in practical terms, with incorrectly thinking that a child is a parent and vice versa. More training data and augmentation may help find the distinctions between family members and reduce this confusion.

Athena Test Sample

LENET-5 (90.72% Sure is Stranger - INCORRECT):

[[6.8331152e-02 1.5640017e-02 7.0141692e-07 9.0729171e-01 8.7364623e-03]]

REGULAR-CNN (88.59% Sure is Teresa - INCORRECT):

[[9.1286443e-02 1.6097942e-03 6.2905085e-05 2.1187244e-02 8.8585359e-01]]

Costaki Test Sample

LENET-5 (39.39% Sure is Stranger - INCORRECT):

[[0.17369762 0.12877548 0.00329115 0.3939123 0.3003235]]

REGULAR-CNN (79.87% Sure is Athena - INCORRECT):

[[7.9868054e-01 1.5272109e-01 1.3584872e-04 4.5567255e-02 2.8952400e-03]]

Costaki Test Sample

```
LENET-5 (94.35% Sure is Stranger - INCORRECT):
[[0.0034588  0.00708388 0.00273169 0.9434956 0.04323001]]
REGULAR-CNN (86.47% Sure is Teresa - INCORRECT):
[[1.8661080e-05 9.0078851e-05 1.8717490e-05 1.3517237e-01 8.6470020e-01]]
```

References

Project repository: https://github.com/Costaki33/COE-379L-Projects/tree/main/project4
How to Build a Deep Facial Recognition App Series by Nicholas Renotte: https://www.youtube.com/watch?v=sQpPaW17TwU&ab_channel=NicholasRenotte
Labeled Faces in the Wild Dataset: http://vis-www.cs.umass.edu/lfw/
Class webpage: https://coe-379l-sp24.readthedocs.io/en/latest/index.html
Class repository: https://github.com/joestubbs/coe379L-sp24/tree/master
ChatGPT: https://chat.openai.com/