Developing a CNN for Identifying Specific Individuals in Varying Image Qualities

Anthony DiBenedetto, Austin Kind

DATA-49000
Department of Engineering, Computing, and Mathematical Sciences
Lewis University
24 November 2024

Abstract

Facial recognition technology has become a vital tool in many applications, from smartphone user authentication to high-level security systems. Through the development of deep learning models, facial recognition technology has significantly improved, especially in recent years. However, identifying individuals across various sets of images remains challenging due to cameras' varying image qualities. While many deep learning methods can effectively handle facial recognition tasks, Convolutional Neural Networks (CNNs) are the most widely used because they excel at extracting key features from images and performing localization tasks [7,9]. This study investigates the use of a CNN to identify Tony Soprano from *The Sopranos* in images of differing noise levels and quality. One of the central challenges of this study is getting the model to work well on images affected by issues such as low resolution, noise, and resizing. To handle this, we added a special layer that can reduce noise, which helps to improve image quality before the model processes it. Previous studies, such as the Noise Gating Dynamic Convolutional Network (NGDCNet), have explored integrating dynamic denoising directly within CNN architectures for image recognition tasks. Their study focused their dynamic denoising convolutions on spatial domain filtering and other methods, such as Block Matching 3D [1]. Our approach also used a dynamic convolutional layer but we utilize a Gaussian filter for image smoothing. Our model will directly smooth pixel intensity variations before feeding it into the rest of the model's layers. This will enhance recognition accuracy specifically for images with varying quality and simulated infrared conditions. This step ensures that the CNN receives cleaner data to work with, enhancing its recognition capabilities. To train the model, a dataset was used containing images of varying levels of noise—low, moderate, high–and infrared simulated images. The images were then annotated using LabelImg. The TensorFlow framework built and optimized the CNN, leveraging its robust tools for deep learning tasks. The model was trained and tested across these conditions to see how well it could maintain accuracy, achieving an overall average accuracy of 72%. Most notably, the model performed best on the base images, which were the images without any modifications, achieving an accuracy of 85.7%. However, the model's performance declined when the target individual appeared smaller within the frame or when the images contained higher levels of noise, highlighting the model's limitations in dealing with extreme resizing, distant subjects, or noise-heavy images. This approach could be useful in fields like security, forensics, and device authentication, where images are not always perfect. While the initial results are promising, the model will still need further refinement in the future to achieve our desired results. Future work will focus on implementing additional techniques like multi-scale detection and training the model on a more extensive dataset.

Acknowledgements

We would like to express our gratitude to the Data Science Department for their support and for equipping us with the essential skills and knowledge that were instrumental in conducting this research successfully. We are especially grateful to Dr. Udagedara for her mentorship and guidance. Finally, we express our thanks to Lewis University for providing us with the opportunity to do this work.

Table of Contents

List of Figures

1. Introduction

1.1 Background

Convolutional Neural Networks (CNNs) are machine learning models that excel at processing images [7]. They work by detecting patterns in data, starting with basic features like edges or lines and building up to more complex shapes and objects. Facial recognition, one of their most common uses, involves identifying or verifying individuals based on their facial features. This technology is widely used in security, surveillance, and even everyday devices like smartphones [2].

However, CNNs often require high-quality images to perform well. In real-world situations, images are often noisy, blurry, or low resolution [6]. Noise refers to unwanted visual distortions, like grainy patches or color inconsistencies, which make it harder for the model to recognize key features. Other challenges include the size of the person's face, where a person in the image appears too small, or low-light conditions, which can further degrade image quality. These problems limit how CNNs can be applied in practical situations like video surveillance or forensic analysis [3]. This study focuses on addressing these challenges using a novel model design.

1.2 Problem Statement

Facial recognition models struggle when dealing with poor-quality images. Current methods usually rely on preprocessing steps to clean up images before feeding them into a CNN. While this can help, it adds complexity and may not work well in real-time applications [6]. Moreover, few models are designed to handle a mix of challenges, such as noise and infrared images [3].

This project addresses these issues by directly integrating a dynamic denoising step into the CNN. The goal is to allow the model to clean up noisy images on the fly, making it more reliable and efficient. The project also tests the model on simulated infrared images, miming thermal imaging used in low-light environments [3]. This approach provides a new way to streamline noise handling and processing, minimizing external dependencies.

1.3 Objectives

The primary objectives of this study are to:

1. Build a dataset of images with varying levels of noise, including simulated infrared images.
2. Design a CNN with a dynamic denoising layer that can handle noisy and low-quality inputs.
3. Test the model’s performance across different conditions and evaluate its reliability.

2. Literature review

2.1 Existing Models

CNNs are the standard choice for tasks involving image recognition and facial detection. They are effective at learning patterns in data without needing manual intervention [2]. However, most CNNs struggle with noisy or degraded images. Researchers have tried to solve this by using noise-reduction techniques. Zhu et al. introduced a CNN that adapts to noise dynamically, improving its ability to handle distorted inputs [1]. While their approach worked for general image tasks, it wasn’t tested on facial recognition in complex settings. Another study by Momeny focused on making CNNs more robust to noise but didn’t explore other challenges like resizing or low-light conditions [8]. Infrared imaging, often used in low-light scenarios, has also been studied. Lin et al. developed a model for thermal face recognition, but it relied on specialized cameras that were expensive and not widely available [3]. Other researchers have explored denoising methods for infrared images, but these are often applied as separate preprocessing steps rather than being integrated into the CNN itself [6].

2.2 Gaps In Knowledge

Despite progress, most models focus on specific challenges like noise or low-light conditions rather than combining them [3]. There is also little research on integrating dynamic noise reduction directly into the CNN architecture. This project aims to address these gaps by building a model that can handle a wide range of image issues, including noise and infrared effects, while simultaneously building a dynamic denoising layer. This altogether has not been tried before.

3. Methodology

3.1 Dataset Preparation

The dataset was created using images of Tony Soprano from the TV show *The Sopranos*. These images were split into five groups: base, low-noise, moderate-noise, high-noise, and simulated infrared [4]. In this context, base images refer to clean, high-resolution images without added distortions, serving as an ideal reference. Low-noise images were clean and sharp, showing clear features without distortion. Moderate- and high-noise images were changed by adding grain and blurriness to make them look more like real-world problems, such as inadequate lighting or low-quality cameras. The major distinction between these noise levels comes from the amount of variance that was added to the images within the Gaussian noise function. The function was designed to take in the image types's variance level and calculate a value σ by taking the square root of the variance level before applying the noise to the image. More specifically, low-noise images had a variance of 0.01, moderate-noise images had a variance of 0.05, and high-noise images had a variance of 0.1. These values were arbitrarily chosen based on previous work we have done in this field. We measured noise levels by checking how much pixel brightness changed in the images [6]. Low-noise images had minor changes in brightness, while higher-noise images had more prominent and uneven changes. This made the higher-noise images look much grainier and more challenging to see clearly. More noise in an image makes details like the edges of a face harder to recognize. For example, in a high-noise image, the face might blend into the background, making it difficult for the model to detect the person [12]. This allowed us to test how well the model can handle tough conditions. We changed the colors for infrared images by boosting the red channel and lowering the green and blue. This made the images look like thermal camera outputs, often used at night or in the dark [3].

Fig. 1: Example of Annotated Images

Each image was labeled using a tool called LabelImg [16]. This tool lets us draw boxes around Tony Soprano's face to show exactly where it was in the image. These boxes, called bounding boxes, were saved as files with the box coordinates. We converted these files into a format that worked with our model, ensuring everything was ready for training. We collected a total of 125 images. Each group had 25 images: the base group, low-noise, moderate-noise, high-noise, and infrared. In this context, base images refer to clean, high-resolution images without added distortions, serving as an ideal reference. Keeping

the number of images balanced across the groups ensured the model could learn equally from all conditions. Figure 1 provides an example of the five image types, with the bounding boxes around Tony Soprano's face. These variations allowed us to test how well the model works with different challenges.

3.2 Model Design

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 512, 512, 1)	0
lambda (Lambda)	(None, 512, 512, 1)	0
conv2d (Conv2D)	(None, 512, 512, 32)	320
max_pooling2d (MaxPooling2D)	(None, 256, 256, 32)	0
conv2d_1 (Conv2D)	(None, 256, 256, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 128, 128, 64)	0
conv2d_2 (Conv2D)	(None, 128, 128, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 64, 64, 128)	0
conv2d_3 (Conv2D)	(None, 64, 64, 256)	295,168
max_pooling2d_3 (MaxPooling2D)	(None, 32, 32, 256)	0
conv2d_4 (Conv2D)	(None, 32, 32, 512)	1,180,160
max_pooling2d_4 (MaxPooling2D)	(None, 16, 16, 512)	0
conv2d_5 (Conv2D)	(None, 16, 16, 512)	2,359,808
max_pooling2d_5 (MaxPooling2D)	(None, 8, 8, 512)	0
conv2d_6 (Conv2D)	(None, 8, 8, 1024)	4,719,616
max_pooling2d_6 (MaxPooling2D)	(None, 4, 4, 1024)	0
dropout (Dropout)	(None, 4, 4, 1024)	0
flatten (Flatten)	(None, 16384)	0
dense (Dense)	(None, 1024)	16,778,240
dense_1 (Dense)	(None, 512)	524,800
dense_2 (Dense)	(None, 256)	131,328
dense_3 (Dense)	(None, 4)	1,028

Fig. 2: Summary of CNN Architecture

The CNN was designed with the specific challenges of noisy and low-quality images in mind. Its architecture included a dynamic denoising layer that cleaned up input images during processing. This layer used a Gaussian filter to smooth out noise, such as grain or pixel inconsistencies while preserving key features like edges and shapes. The cleaned-up images were then passed through a series of seven convolutional layers and seven pooling layers. These layers extracted features like patterns and facial structures, which are essential for identifying the subject [1]. The deeper layers of the CNN consisted of fully connected nodes that combined the extracted features to predict bounding box coordinates. The final output layer returned four numerical values: xmin, ymin, xmax, and ymax. This defined the location of the bounding box around Tony Soprano's face.

The model uses max pooling layers to reduce the spatial dimensions of feature maps, improving computational efficiency. This reduction was critical for quickly processing images while focusing on significant features. Additionally, the architecture used a dropout layer with a rate of 0.4 to prevent overfitting by randomly deactivating some neurons during training. This ensured the model could generalize effectively to unseen data, particularly given the variability in noisy images.

Four fully connected dense layers at the end combined all the extracted features to finalize the bounding box predictions accurately. The model was designed in TensorFlow [14], which made it simple

by adding custom layers like the dynamic denoising layer. The architecture prioritized efficiency while maintaining quality across varying noise levels and image conditions [2]. Figure 2 summarizes the model's architecture, including six convolutional blocks, pooling layers, and a dropout layer to prevent overfitting.

The final output layer predicts the four bounding box coordinates. The dynamic denoising layer, implemented using TensorFlow operations, calculated image variance and applied a smoothing filter when noise exceeded a threshold. This layer dynamically adapted to the noise level, ensuring only necessary smoothing was applied to avoid over-blurring fine details. The threshold, set to 80% of the calculated variance, allowed the layer to retain sharp edges while cleaning noisy pixels, ensuring high-quality inputs for the model. This balance between performance and precision was crucial for real-world noisy images, where both clarity and computational efficiency are vital.

3.3 Training and Testing

Once the dataset was ready and the CNN was designed, the training and testing process began. The dataset was divided into two parts. The first is 80% for training, and the second is 20% for testing. This split ensured the model had enough data to learn while leaving some of the data untouched for evaluation [9]. Data augmentation techniques were applied during training to make the model more resilient to different scenarios. These techniques included flipping the images horizontally, rotating them slightly, and adjusting their brightness. These transformations increased the variety of the training data and helped the model generalize better to unseen images [11]. This can also be interpreted as new images in the model, as it views the same images from new angles. The model was trained using the Adam optimizer, which is effective for adjusting weights in deep-learning models [6]. The loss function used was mean squared error (MSE), as seen in Figure 3, which measured the difference between the predicted and actual bounding box coordinates, by taking the average of the squares of the errors. The training involved running the model through multiple epochs, during which it adjusted its internal parameters to minimize the error [10]. To evaluate the model, we used the testing set. Mean absolute error (MAE) was the primary metric we used. This was calculated to measure how accurately the model predicted bounding boxes for noisy and infrared images. The results showed that the dynamic denoising layer significantly improved performance on noisy data, although challenges remained when the area of the bounding box containing the individual's face was smaller in comparison to others [12]. We also introduced our own custom metric to track the accuracy of binary classification. From the bounding box predicted on the test set by the model, we would take the difference between the predicted bounding box and the ground truth bounding box. Then we would take the absolute value of this difference and then compare it with a set threshold of 80 pixels. If the collective error was above 80, we would classify the prediction as a failure. If it was below 80, we would classify it as a successful classification. We originally began by arbitrarily choosing a threshold of 100 pixels and worked our way down until the results of the model drastically began to decrease, which is how we arrived at our current pixel threshold.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Fig. 3: Mean Squared Error Equation

4. Results

4.1 Performance Metrics

The model showed strong performance on low-noise images, with decent accuracy in predicting bounding boxes. Moderate-noise images posed some challenges, but the dynamic denoising layer improved the model's performance significantly compared to a baseline without noise reduction. High-noise images and simulated infrared images were the most difficult for the model. In these cases, accuracy dropped, especially when the subject's head appeared small in the frame. The model's predictions on high-noise images were inconsistent. When the noise severely changed facial features, the model often failed to locate the subject accurately. *Key findings:*

- Low-noise images: High accuracy, with bounding boxes closely matching the ground truth.
- Moderate-noise images: Reasonable performance but occasional misalignment in predictions.
- High-noise images: Significant drop in accuracy; bounding boxes were often misplaced or too large.
- Infrared images: Performance was acceptable but less reliable, with some bounding boxes missing the subject entirely.

4.2 Visualization

Fig. 4: Example of CNN Prediction Bounding Box Results

Figure 4 provides an example of the model's predicted bounding box in blue, compared to the actual bounding boxes for the different image types in green. The model performs well on base and little-noise images but shows inconsistencies for high-noise and moderate-noise conditions, as shown above.

5. Discussion

5.1 Implications

The denoising layer helped the model maintain accuracy for low and moderate-noise images, showing that this approach can address certain real-world challenges. However, the drop in performance on high-noise and infrared images points to areas where the model needs further improvement, which can be something we work on in the future. This project is a solid base for this work and won't be difficult to add and expand on. This suggests that additional strategies, such as multi-scale detection or more advanced noise reduction techniques, should be tried in the future. Another direction to explore is making the model more diverse by training it alongside pre-trained face models. By teaming it up with existing models that recognize a wide variety of faces, we could create a dynamic denoising system that works effectively for all faces, not just a single individual. This could help broaden its application, making it useful in settings like public surveillance or general-purpose facial recognition systems.

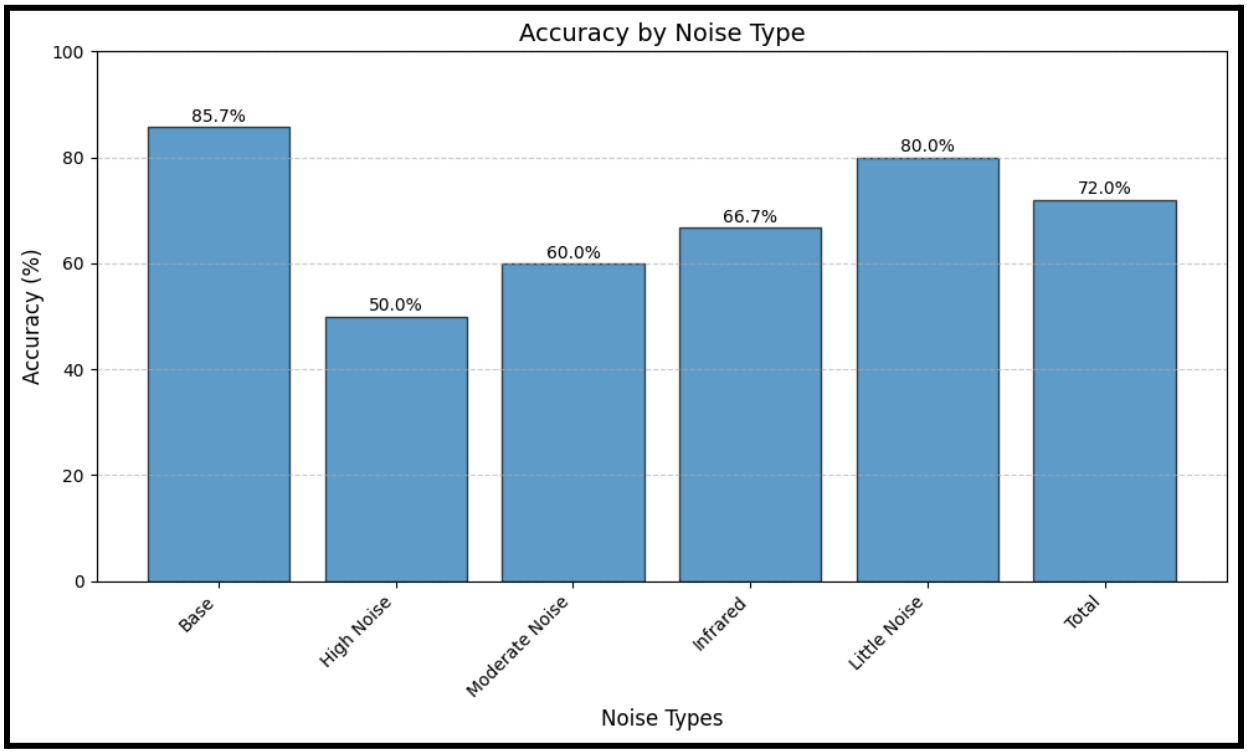

Fig. 5: Accuracy of CNN across Varying Noise Types

Figure 5 displays the model's accuracy for each of the varying image types, with the total column representing the average accuracy across all types. Figure 5 shows that the model obtained results that match the trend we had hoped for. One thing to keep in mind is that our error threshold is set to 80 pixels. The smaller our threshold, the worse our results become, as there is less room for error. We hope that even with an allowed error of 80 pixels, we can see trends in our model and build off of them. In the future, we want to progressively lower the allotted error from 80 to 20 pixels. Our base group, which is the original images unaffected by noise or quality, achieved an accuracy of 85.7% under these conditions, which is expected as no augmentation was made. One of our most significant findings is that the high noise obtained 50% accuracy using our dynamic denoiser. While 50% is not a very high accuracy, this does show that the model is still performing relatively well under the most grainy circumstances in its early stage.

5.2 Ethical Considerations

This study was conducted with careful attention to ethical concerns. The dataset was created using publicly available images from *The Sopranos*, ensuring that no private or sensitive data was used since these have clear licensing and usage rights.

6. Conclusion

This project explored the integration of dynamic denoising into a CNN for facial recognition under challenging conditions. While the model performed well on low- and moderate-noise images, its accuracy decreased on high-noise and infrared images, most notably when the subject's head appeared small. These results show the need for further changes. In comparison to other models our results showed that our model is building in the right directions. Many other models use multiple different denoising techniques within their model, but we stuck with just one, Gaussian smoothing. We showed that Gaussian

smoothing is very powerful and can be used in a dynamic convolution. When it is applied on a variety of different images alone it proves to be very effective. Future work will focus on addressing these limitations. Techniques like multi-scale detection, additional data augmentation, and training on a larger dataset could improve the model's performance. This project serves as a starting point, and there are plans to continue this research in the upcoming semester.

References

[1] M. Zhu and Z. Li, "NGDCNet: Noise Gating Dynamic Convolutional Network for Image Denoising," *Electronics*, vol. 12, no. 24, p. 5019, 2023. Available: https://doi.org/10.3390/electronics12245019. Accessed: Sep. 15, 2024.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. Available: https://dl.acm.org/doi/pdf/10.1145/3065386. Accessed: Sep. 15, 2024.

[3] S. D. Lin, L. Chen, and W. Chen, "Thermal face recognition under different conditions," *BMC Bioinformatics*, vol. 22, no. 313, 2021. Available: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-021-04228-y. Accessed: Sep. 15, 2024.

[4] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018. Available: https://arxiv.org/pdf/1804.02767. Accessed: Sep. 15, 2024.

[5] A. Signoroni, M. Savardi, A. Baronio, and S. Benini, "Deep Learning Meets Hyperspectral Image Analysis: A Multidisciplinary Review," *Journal of Imaging*, vol. 5, no. 5, p. 52, 2019. Available: https://www.mdpi.com/2313-433X/5/5/52. Accessed: Sep. 15, 2024.

[6] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, no. 7, 2019. Available: https://vciba.springeropen.com/articles/10.1186/s42492-019-0016-7. Accessed: Sep. 15, 2024.

[7] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014. Available: https://arxiv.org/pdf/1409.1556. Accessed: Sep. 15, 2024.

[8] M. Momeny, A. M. Latif, M. A. Sarram, R. Sheikhpour, and Y. D. Zhang, "A noise robust convolutional neural network for image classification," Available: https://www.sciencedirect.com/science/article/pii/S2590123021000268. Accessed: Sep. 15, 2024.A noise robust convolutional neural network for image classification - ScienceDirect

[9] K. Ahmad, J. Khan, and M. S. U. D. Iqbal, "Convolutional Neural Networks for Noise Classification and Denoising of Images," IEEE, Available: https://ieeexplore.ieee.org/document/8929277. Accessed: Sep. 15, 2024.

[10] D. Sil, A. Dutta, and A. Chandra, "A Comparative Study of Different Denoising Techniques in Digital Image Processing," IEEE, Available: https://ieeexplore.ieee.org/document/8880389. Accessed: Sep. 15, 2024.

[11] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," Visual Computing for Industry, Biomedicine, and Art, vol. 2, p. 7, 2019. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7099553/. Accessed: Sep. 15, 2024.

[12] R. Abiko and M. Ikehara, "Blind Denoising of Mixed Gaussian-Impulse Noise by Single CNN," IEEE, Available: https://ieeexplore.ieee.org/abstract/document/8683878. Accessed: Sep. 15, 2024.

[13] S. M. A. Sharif, R. A. Naqvi, and M. Biswas, "Learning Medical Image Denoising with Deep Dynamic Residual Attention Network," Mathematics, vol. 8, no. 12, p. 2192, 2020. Available: https://www.mdpi.com/2227-7390/8/12/2192. Accessed: Sep. 15, 2024.

[14] "TensorFlow," TensorFlow.

[15] "NumPy," NumPy.

[16] "labelImg 1.8.6," Maintainers: Tzutalin.

Appendix

1. Code

The full code for this project can be found at the following links:

[https://github.com/adibenedetto117/DATA-Capstone/blob/main/prog_1.ipynb]

This code preprocesses our data, ensuring that all of the images are the same size, 512x512. In addition, this code applies varying levels of Gaussian noise to the images.

[https://github.com/adibenedetto117/DATA-Capstone/blob/main/prog_2.ipynb]

This code takes all of our images and creates new versions of them with a simulated infrared. Also in this code you will find our initial creation for our CNN model.

[https://github.com/adibenedetto117/DATA-Capstone/blob/main/prog_3.ipynb]

This code built on top of the CNN model previously created and added our custom dynamic denoising layer. This code also showed some results based on our models predictions.

[https://github.com/adibenedetto117/DATA-Capstone/blob/main/prog_4.ipynb]

This code built on our CNN model and custom denoising layer made previously by adding more convolution layers, a custom call back function, and implementing a custom success rate. Also we refined our dynamic denoising model by having the gaussian filter apply a variable amount of smoothing.