Simultaneous Multi-Face Detection in Real-Time Video Streams

Akhil Tadiparthi and Ishika Patel

University of Colorado CSCI 5922 - Spring 2024

Abstract. In the realm of digital surveillance and user interaction technologies, the accurate and efficient detection of multiple faces in real-time video streams has become increasingly important. This paper explores an innovative approach to multi-face detection in live video by integrating deep learning with advanced image augmentation techniques. We developed a novel dataset simulating real-world webcam scenarios and trained convolutional neural networks (CNNs) on this enhanced dataset. Our approach not only improves facial recognition accuracy in dynamic environments but also addresses the challenges posed by diverse conditions such as varying light, occlusions, and movement, commonly referred to as ”in the wild” scenarios.

Keywords: Image Augmentation, Real-Time Facial Recognition

1 Introduction

1.1 Motivation of Work

In the era of digital surveillance, security systems, and smart interaction technologies, the ability to accurately and efficiently detect multiple faces in live video feeds is paramount. With approximately 51 million households equipped with video surveillance systems and 37% adorned with video doorbells to unlock your banking app with your face and monitor at self-checkouts, these systems underscore the growing demand for enhanced security measures and seamless user experiences. Central to this paradigm shift is the imperative for accurate detection and recognition of multiple faces in real-time video feeds. The motivation of our work is to meet the demand for a robust, real-time multi-face recognition system with improved facial recognition confidence in complex, dynamic scenes. We approach a solution by exploring how improved and altered image data fit real-world webcam scenarios.

1.2 Existing Work

”A Survey on Face Detection in the Wild: Past, Present, and Future” by Zhang et al. (2019) is a comprehensive review paper that discusses the landscape of face detection techniques. This paper refers to the colloquial term, ”in the wild" settings, denoting uncontrolled environments like varying lighting conditions, occlusions, and pose variations. Existing facial recognition solutions leverage algorithms to draw similarities between images to output labels. Popular algorithms are Local Binary Pattern Histograms and Haar Cascade Classifier. These sorts of settings pose significant challenges to face recognition algorithms, seeping into the spatial recognition space. Facial recognition technologies available leverage algorithms to draw similarities between images to output labels.

1.3 Proposition

In this paper, we propose a novel approach to live face recognition that leverages deep learning algorithms optimized for real-time, multi-face detection and recognition. Our solution introduces a unique combination of convolutional neural networks (CNNs) and image augmentation techniques to improve detection accuracy across a variety of challenging conditions outlined as "in the wild". This approach is designed to efficiently process video streams in real-time, accurately recognizing multiple faces simultaneously with accuracy. For this, we created a novel image dataset a model can be trained on that reflects real-world webcam input.

2 Related Work

2.1 Deep Learning in Face Detection

1. Zhang, Y., et al. (2019). "A Survey on Face Detection in the Wild: Past, Present, and Future." Computer Vision and Image Understanding.
2. Li, H., et al. (2020). "Deep Convolutional Networks for Real-Time Face Detection." IEEE Transactions on Image Processing.
3. Wang, X., et al. (2021). "Multi-Cascade Convolutional Neural Networks for Robust Face Detection." International Journal of Computer Vision.
4. Chen, L., et al. (2018). "Efficient and Accurate Face Detection Using Sparse CNNs." CVPR.

Our work is distinct from these studies as it leverages image augmentation methods to improve real-time performance in dynamic environments, unlike the approaches so far that focus on static image analysis. In these studies, typical images in train and test sets leverage a front-on facial view without much emphasis on dynamic angles, lighting, etc. Our methodology is specifically designed to enhance consistency and detection robustness in video streams, addressing challenges posed by fast-moving subjects and varying external factors. However, taking from popular work, we leverage a convolutions neural network for facial recognition.

2.2 Image Augmentation and Algorithms for Face Recognition

1. Lv, J., et al. (2017). ”Data augmentation for face recognition.” Neurocomputing.
2. Masi, I., et al. (2019). ”Face-Specific Data Augmentation for Unconstrained Face Recognition.” International Journal of Computer Vision.
3. Zhuchkov, A. (2021). ”Analyzing the Effectiveness of Image Augmentations for Face Recognition from Limited Data.” 2021 International Conference ”Nonlinearity, Information and Robotics” (NIR).

Our approach differs by specifically creating an ultimate mesh of these popular image augmentation algorithms to be integrated with a model that engages in the real-time recognition scenario. Our integrated model emphasizes enhancing recognition accuracy for ”in the wild” settings by leveraging different image augmentation techniques on our datasets. Our work focuses on using multiple image augmentation techniques and understanding how each impacts CNN model prediction confidence.

3 Methods

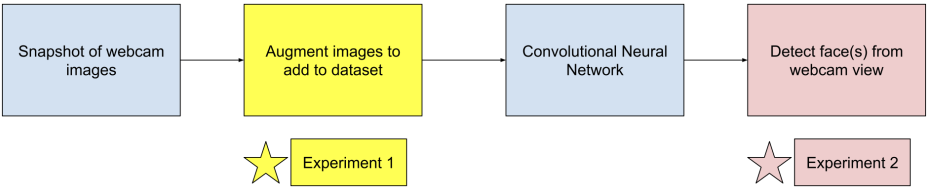
Snapshot of webcam images
Augment images to add to dataset
Convolutional Neural Network
Detect face(s) from webcam view
Experiment 1
Experiment 2

Fig. 1. Experiment architecture.

The figure above outlines the architecture from collecting live images, processing these images using chosen image augmentation techniques, and testing in live facial recognition.

3.1 Step 1: Data Collection

The dataset used was crafted from snapshots taken from a webcam view. Data collection was from a webcam using the OpenFace face detector to collect 200 grayscale images for each user. The Open Face pre-trained detector finds a user’s face and takes consecutive images given the bounding box of the face. To process the data we resized the images to a uniform area of 224x224 pixels and converted the images to gray scale.

3.2 Step 2: Image Augmentation Process

Experiments 1 and 2 focus on developing and determining the accuracy of image augmentation algorithms. Four techniques were chosen to individually augment the images and applied to the dataset:

Rotation
Rotation was implemented to simulate the ”in the wild” setting of head tilting. We implemented a rotation function passing a PIL image of a face along with the desired rotation angle. The function internally employs the ”rotate” method from the PIL library to perform the rotation and returns the resulting rotated face image. We chose to create data given a 10-degree rotation to simulate a head tilt in the wild. This augmentation was considered with potential to improve live-feed recognition with users shifting and moving.

Adjust contrast and brightness
Adjustment of contrast and brightness was implemented to simulate data from overexposed lighting settings. With our implementation, we provide a PIL image along with the desired brightness and contrast adjustment factors. The function internally utilizes the ImageEnhance module from the PIL library to separately enhance the brightness and contrast of the image based on the provided factors, returning the resulting image with adjusted brightness and contrast. We chose to adjust the contrast by 2x and brightness by 3x on each image.

Random Erasing
Random erasing was implemented to simulate ”in the wild” image occlusion scenarios. A PIL image is required as input along with optional parameters controlling the probability and characteristics of the random erasing process: Probability of applying random erasing, minimum proportion of erased area, and maximum proportion of erased area. Initially, the PIL image is converted into a numpy array to facilitate manipulation. Then calculate the dimensions of the erasing rectangle based on the specified range of proportions and aspect ratios. A random position within the image is then chosen for the erasing rectangle. Finally, the function modifies the numpy array by filling the designated rectangle with random pixel values between 0 and 224, simulating the effect of erasing. The resulting modified numpy array is converted back into a PIL image, which is then returned as the output of the function, reflecting the original image with the random erasing applied.

For our image augmentation using random erasing, we applied random erasing to 100% of the images taken on the webcam. A random selection between 20-40% of each image was erased to simulate occluded and obscured views.

Gaussian Filter
A Gaussian filter was implemented to simulate noise and rapid movement in

real-time video streams. A PIL image along with parameters alpha to control the scaling factor for the deformation field and sigma to configure the standard deviation of the Gaussian filter used for smoothing. A random deformation field is generated using the provided parameters and SciPy's gaussian_filter. The deformation field is applied to the input image using SciPy's map_coordinates function, which performs interpolation to compute the intensity values of the distorted image at non-grid locations. The resulting distorted image is returned as a PIL image object with Gaussian transformations applied. In this experiment, we applied the Gaussian Filter with alpha as 50 and sigma as 5.

An example of all the image augmentation algorithms used are shown below:

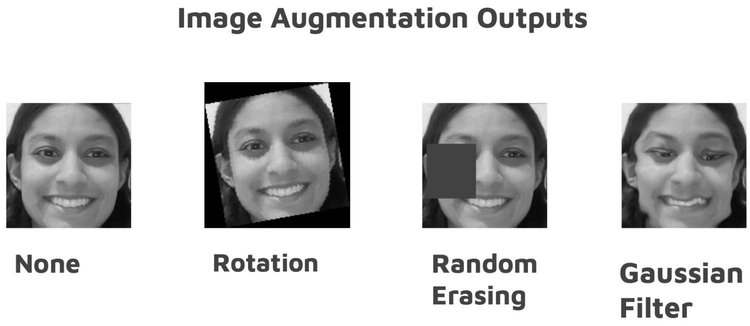

Fig. 2. Image Augmentation Outputs

3.3 Step 3: Convolutional Neural Network

After creating our novel dataset and implementing image augmentation algorithms, we propose a novel CNN to train our datasets on. Our CNN architecture comprises an input layer that receives the prepossessed face images, resized to a uniform dimension of 224 x 224 pixels. In our methodology, we employed CNN architecture consisting of six sets of Conv3D layers, using 256 filters of 3x3 to extract spatial hierarchies of features (edges in the initial layers to complex features in deeper layers). The convolutional layers are interspersed with ReLU activation functions to introduce non-linearity, and max pooling layers to down-sample the feature maps and reduce computation. Lastly, we have the fully connected layer, which collapses the spatial features into a flat vector for classification. That is attached to a softmax activation function, categorizing the faces with the different people. We use cross-entropy for the loss function since we perform classification, a learning rate of 0.001 (default value), and Adam optimizer(default). We trained for 50 epochs, using a batch size of 32.

3.4 Step 4: Getting Results

We then deploy the model on real-time webcam streams, choosing different webcam scenarios such as moving in the frame, different lighting, and obscured face views. We measure the confidence and how it changes as we deploy different models trained on different datasets(individual image augmentation algorithms).

4 Experiments and Results

Experiment 1: Comparison of Image Augmentation Techniques

1. **Main Purpose:** Compare our image augmentation techniques to fine-tune a new image augmentation process to use in real-time facial detection.
2. **Evaluation Metric:** Confidence in prediction.

For this experiment we created separate datasets (that we trained multiple models on, one for each dataset) for each augmentation algorithm, evaluating one algorithm at a time.

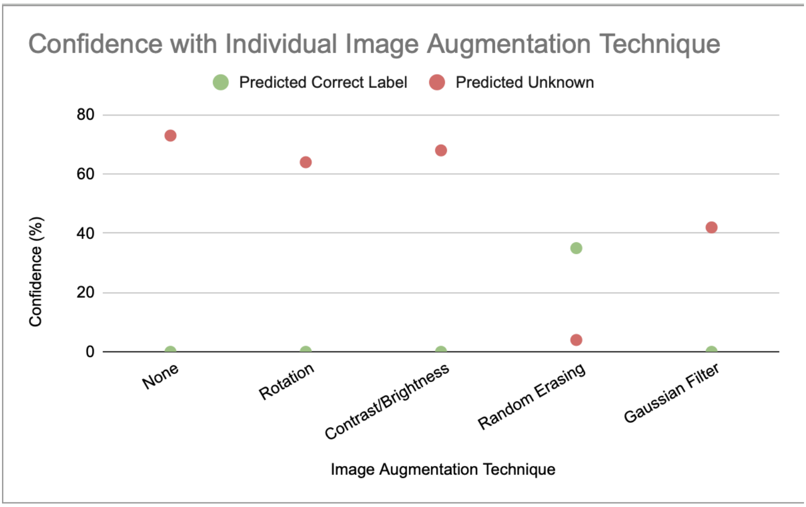

Fig. 3. Experiment 1 Results: One augmentation algorithm at a time

The Predicted Correct Label shows the confidence the model recognizes the face correctly, while the Predicted Unknown shows the confidence of it being unknown. As shown in the figure above, individually, Random Erasing was the most effective with the confidence of correct predictions close to 40% and unknown predictions close to 0%. The remaining four augmentation algorithms individually had the confidence of correct predictions close to 0% and unknown predictions in the range of 40-75%.

Experiment 2: Successive Addition of Image Augmentation Techniques to Create a High-Performing Model

1. **Main Purpose:** Evaluate the model's performance in terms of confidence in prediction as we consecutively add new image augmentation data points to the dataset.
2. **Evaluation Metric:** Confidence in prediction.

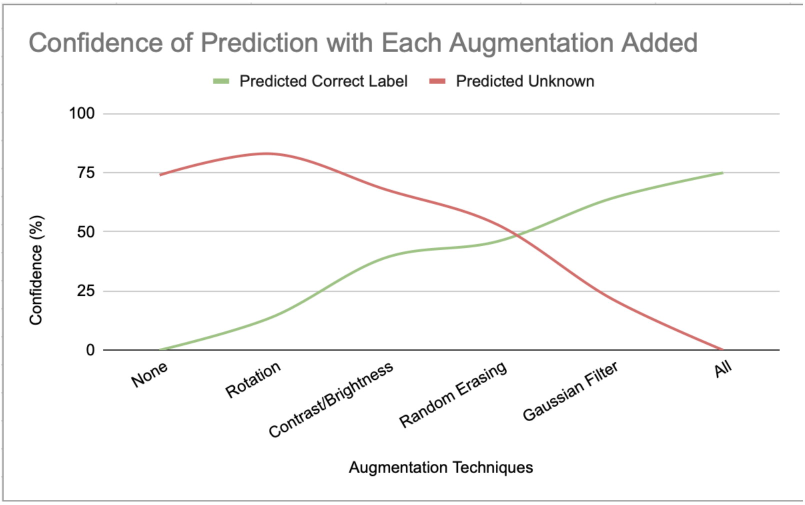

Fig. 4. Experiment 2 Results: Successive augmentations added

This results chart shows where we trained a model on data that was consolidated with each successive augmentation algorithm applied. For example, None, and then None & Rotation, and then None & Rotation & Contrast/Brightness until there is a model trained on all original images and all images generated with the image augmentations. The ordering of these successive additions is from least complex to most complex. This is so that we can observe how an augmentation algorithm's complexity can result in higher evaluation performance. As before, the Predicted Correct Label shows the confidence the model recognizes the face correctly, while the Predicted Unknown shows the confidence of it being unknown.

From this chart, we see that as we add more image augmentation algorithms, the model increases in confidence predicting the correct face. We started at approximately 75% of an unknown prediction and moved it down to 0% with all the augmentation techniques together. On the other hand, the predicted correct label confidence went from 0% to 75% with all the augmentation techniques together.

5 Downfalls & Future Research

One observation as to why the results are turning out the way they are is due to dataset sizes. Since we are successively adding augmented images in Experiment 2, the dataset that the model is trained on is increasing with every augmentation. This results in higher accuracy due to more data that the model is being trained on. This can sometimes not be as comparable to Experiment 1 since that is working with individual datasets of the same size.

To further enhance the performance and applicability of our system, we propose several future directions. To address the disparities in model training due to dataset size variations observed in our experiments, future work will focus on creating balanced datasets where all image augmentations are equally represented. This will ensure consistent training conditions and more reliable comparisons across different augmentation techniques. We can also explore the integration of additional machine learning algorithms and advanced neural network architectures that could further boost the robustness and accuracy of face recognition systems. For example, techniques such as generative adversarial networks (GANs) for more sophisticated image augmentations might be explored. Lastly, improving the computational efficiency to support faster processing speeds will be crucial for deployment in real-time systems. Specifically, we can look into optimizing the current CNN model to reduce latency without compromising accuracy.

6 Conclusion

6.1 Key Findings

Our work focused on enhancing in-the-wild recognition by developing a unique image dataset derived from real-world webcam inputs, aiming to tackle associated challenges. Through training a deep learning model on all four selected image augmentation techniques, we observed the highest confidence in real-time face recognition. Interestingly, when tested independently, the random erasing technique emerged as particularly effective in bolstering confidence levels in recognition tasks. This underscores the significance of both comprehensive training data and innovative augmentation methods in improving the robustness of recognition systems for real-world applications.

6.2 Ethics

In addition to technical advancements, our project raises important ethical considerations surrounding facial recognition and data augmentation techniques. Utilizing live data to train models introduces complexities regarding consent and privacy, especially when sourced from public settings where individuals may not be aware their images are being captured and used. Moreover, the deployment of facial recognition systems can exacerbate existing biases and disparities if not carefully monitored and mitigated, potentially leading to unfair treatment or discrimination against certain demographic groups.

7 References

1. Chen, L., et al. (2018). ”Efficient and Accurate Face Detection Using Sparse CNNs.” CVPR.
2. Kumar, S., et al. (2020). ”Lightweight Deep Learning Model for Mobile Real-Time Face Detection System.” Pattern Recognition Letters.
3. Li, H., et al. (2020). ”Deep Convolutional Networks for Real-Time Face Detection.” IEEE Transactions on Image Processing.
4. Lv, J., et al. (2017). ”Data augmentation for face recognition.” Neurocomputing.
5. Masi, I., et al. (2019). ”Face-Specific Data Augmentation for Unconstrained Face Recognition.” International Journal of Computer Vision.
6. Wang, X., et al. (2021). ”Multi-Cascade Convolutional Neural Networks for Robust Face Detection.” International Journal of Computer Vision.
7. Yang, S., Luo, P., Loy, C. C., and Tang, X. (2016). WIDER FACE: A Face Detection Benchmark. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
8. Zhang, Y., et al. (2019). ”A Survey on Face Detection in the Wild: Past, Present, and Future.” Computer Vision and Image Understanding.
9. Zhuchkov, A. (2021). ”Analyzing the Effectiveness of Image Augmentations for Face Recognition from Limited Data.” 2021 International Conference ”Nonlinearity, Information and Robotics” (NIR).
10. PIL Python Package
11. Scipy Python Package
12. Numpy Python Package
13. Open Face