Evaluating Various Optimization Techniques on Live Face Captured Images

Nickolaus White, Aditya Jain, Mark Carravallah, Stephen Lee, Hunter Pierson
Dept. of Computational Mathematics, Science, and Engineering, Michigan State University
CMSE831: Computational Optimization
Dr. Longxiu Huang
December 7, 2023

Evaluating Various Optimization Techniques on Live Face Captured Images

This project explores the development of a facial recognition system aimed at accurately predicting the identity associated with a given face. Utilizing images collected through a webcam, we employed a diverse set of machine learning models, including Multi-Layer Perceptron (MLP), Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), Logistic Regression, Random Forest, and a Convolutional Neural Network (CNN). The goal was to assess the performance of these models in accurately associating facial features with corresponding identities and then optimizing those models with techniques learned in class to improve accuracy and/or efficiency.

To enhance the efficiency of our models, we incorporated dimensionality reduction techniques such as Principal Component Analysis (PCA) and Robust Principal Component Analysis (RPCA). These techniques were employed to optimize the feature space and reduce computational complexity, thereby improving the overall performance of the facial recognition system.

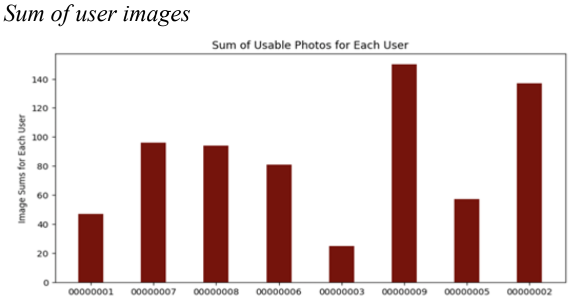
Our project provides insights into the comparative effectiveness of different machine learning algorithms and dimensionality reduction approaches in the context of facial recognition. The findings contribute to the ongoing efforts to create robust and accurate facial recognition systems with potential applications in security, access control, and human-computer interaction.

Data Collection and Exploratory Data Analysis

As we conducted the data collection process personally, the need for exploratory data analysis was minimal. However, we employed two primary strategies. Firstly, we standardized the dimensions of all images, and secondly, we identified viable images, as illustrated in Figure 1 below. As elaborated later, there were certain instances, particularly involving "faces," where we encountered challenges obtaining usable pictures, consequently influencing the final results.

Figure 1

Sum of user images



Note. The above graph represents the sum of images grouped by user ID.

To initiate the development of a live facial recognition system, a diverse set of facial images displaying various expressions is crucial. The images were captured using a Logitech HD Pro C920 Webcam, and a USB2.0 HD UVC functional webcam, operating at dimensions of 640 x 480 pixels. The feature selection dimensions are set to 90-200 x 90-200, and the pixel dimensions are fixed at 150 x 150 using both PCA and RPCA. Additionally, the images undergo random rotation and flipping for enhanced variability.

The image capture process employs the OpenCV library to access the computer's webcam, capturing real-time facial images. The system is designed to rapidly capture 50 consecutive photos after the user inputs an eight-digit User ID and presses the ‘Enter’ key. A rectangular box ensures that the detected face is within the webcam's field of view.

Post-capture, the images are stored in a designated folder, each tagged with a User ID and a unique photo number for subsequent processing. Testing has been conducted over several weeks, predating the CMSE830 course, Computational Optimization. Previous research in applied machine learning, with a partial focus on deep learning algorithms, laid the foundation for this project. Motivated by this research, an updated iteration of the facial recognition system ensued, emphasizing meticulous participant photo selection to address the incomplete aspects of the prior iteration.

Ensuring the robustness of the system involves capturing images under diverse conditions, including different facial expressions, lighting, and backgrounds. The data collection strategy considered various environments, angles, and expressions to comprehensively assess the system's capabilities.

In the trial of the facial recognition system, the dataset was created with myself and seven volunteers, generating approximately 800 images. The relatively lower number of classes is anticipated to yield high prediction accuracy. However, it is acknowledged that results may vary based on image quality and dataset size.

Model Selection/Implementation

We used a variety of models to test the image recognition and compare them to one another. We had preconceived ideas on which would work best but wanted to test them with our own data. For image recognition especially, we had a good idea that the CNN would perform as well as the MLP and SVC classifiers. However, the outcome yielded surprising results from one of our chosen models, an aspect that will be discussed in detail in the forthcoming "Model Results/Conclusion" sections. The subsequent models were utilized, each briefly explained below.

MLP

MLPs are a class of feedforward artificial neural networks. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. MLP is well-suited for facial recognition due to its capacity to automatically learn hierarchical representations of facial features, adapt to complex and non-linear relationships in the data, and provide a versatile framework for effective pattern recognition and classification in high-dimensional image datasets.

SVC

SVC aims to find an optimal hyperplane in a high-dimensional space that effectively separates data points into distinct classes. It identifies support vectors, which are the data points crucial for defining the decision boundary. SVC is well-suited for facial recognition due to its effectiveness in handling high-dimensional data, identifying complex patterns inherent in facial features, and providing a robust classification framework for accurately distinguishing between individuals based on facial characteristics.

Random Forest

Surprisingly our best performing model before we optimized the data with RPCA was Random Forest. It excelled in our facial recognition project due to its ensemble learning, aggregating predictions from multiple decision trees to enhance robustness and generalization. The algorithm's ability to assess feature importance proved valuable for identifying discriminative facial features. Its effectiveness in handling high-dimensional data, robustness to overfitting, tolerance to missing data, and versatility in parameter tuning contributed to its success in optimizing performance for our project.

CNN

Our CNN architecture, designed for facial feature recognition, consists of multiple convolutional layers with max-pooling. The initial layer, with 16 filters of size 3x3, begins feature extraction, followed by layers with increased filters (32 and 16) for detailed feature mapping. The network concludes with fully connected layers for classifying distinct identities. The CNN model's training focused on high accuracy and computational efficiency. Key strategies included:

- Data Augmentation: Implemented random rotations, flips, and cropping to expose the model to varied facial orientations and expressions.
- Learning Rate Scheduling: Employed a dynamic learning rate, decreasing over epochs for precise weight adjustments.
- Early Stopping and Model Checkpointing: Used to prevent overfitting and save the best model state based on validation loss.

We utilized the Hyperband method for hyperparameter optimization, searching for the best configurations in filters, kernel sizes, dense units, and learning rates.The CNN model was rigorously tested for accuracy in facial recognition, effectively distinguishing between individuals. Its robustness was confirmed through cross-validation, ensuring reliable and consistent performance.

Optimization

The optimization technique we relied on was decomposition, specifically Principal Component Analysis (PCA) and Robust Principal Component Analysis (RPCA). PCA is a powerful statistical technique widely used for dimensionality reduction and feature extraction in multivariate data analysis. The fundamental objective of PCA is to transform a possibly correlated set of variables into a new set of uncorrelated variables, known as principal components. These components are linear combinations of the original variables and are ordered by the amount of variance they capture, with the first principal component explaining the maximum variance in the data.

The PCA process begins by computing the covariance matrix of the original dataset. The eigenvectors and eigenvalues of this covariance matrix are then calculated. The eigenvectors represent the directions (principal components) in which the data varies the most, while the corresponding eigenvalues indicate the magnitude of the variance along each principal component. By selecting the top eigenvectors based on their corresponding eigenvalues, one can create a transformation matrix to project the data into a new subspace.

The resulting transformed data, represented by the principal components, often allows for a more compact and efficient representation of the original data. This not only aids in reducing dimensionality but also highlights the most important patterns or structures inherent in the dataset. PCA finds applications in various fields, including image processing, pattern recognition, and machine learning, where it is used to enhance computational efficiency, remove redundant information, and uncover underlying trends in complex datasets. Its versatility and simplicity make PCA a valuable tool for exploratory data analysis and feature engineering in diverse domains.

We also utilized RPCA, which assumes a pivotal role in managing variations and outliers that may manifest in facial images. Facial recognition systems confront challenges arising from diverse lighting conditions, occlusions, and facial expressions, leading to potential data corruption or noise (See Figure 2).

Figure 2

Preview of user images

Note. The above shows the sample images of various users

RPCA addresses these challenges by decomposing facial image datasets into a low-rank component, signifying intrinsic facial features, and a sparse component, capturing outliers or distortions. This segregation enables facial recognition algorithms to concentrate on essential facial characteristics, thereby alleviating the influence of outliers. Consequently, this enhances the robustness and precision of the recognition process, particularly in real-world scenarios where environmental factors can impact image quality. The method involves computing the Frobenius Norm of the sum of the base image (low-rank) and the outliers (sparse), fostering a concise representation of the primary data structure. Additionally, it facilitates the identification and isolation of outliers or sparse errors in the data, allowing for the optimization of the original image.

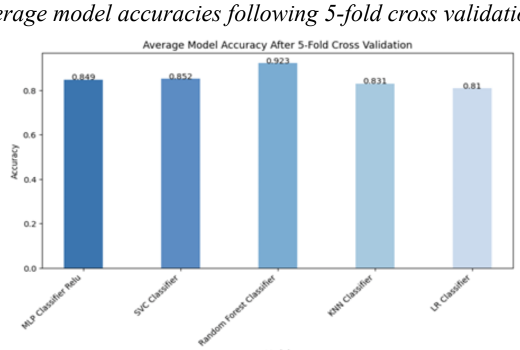
Model Results

Classifier Results

Our expectations regarding model performance were largely met, with the MLP and SVC emerging as the top-performing models, closely followed by KNN. It is noteworthy that, as previously highlighted, the Random Forest model demonstrated superior performance in the initial run without RPCA after 5-fold cross-validation, as depicted in Figure 3. Furthermore, this model maintained the highest accuracy on the test set, as illustrated in Figure 4.

Figure 3

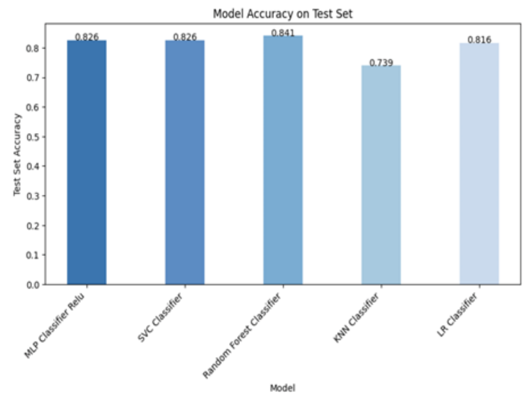
Average model accuracies following 5-fold cross validation



Note. The above figure depicts the average accuracies of each model used after utilizing 5-fold cross validation.

Figure 4

Average model accuracies on test set portion of data w/o RPCA

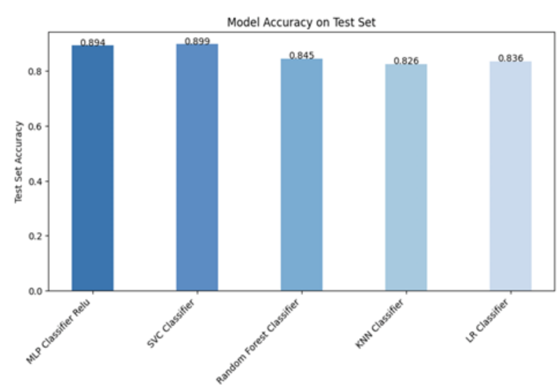


Note. The above figure depicts the average accuracies of each model used on the test data set without RPCA applied.

Following the implementation of the RPCA optimization, a notable improvement in test accuracy was observed for our MLP and SVC models. Contrarily, the performance of our Random Forest model showed a decline compared to its previous test run before implementation of RPCA. This outcome aligns with our initial expectations regarding the success of different classifiers. Comparisons are presented in Figure 5 below.

Figure 5

Average model accuracies on test set portion of data w/ RPCA



Note. The above figure depicts the average accuracies of each model used on the test data set with RPCA applied.

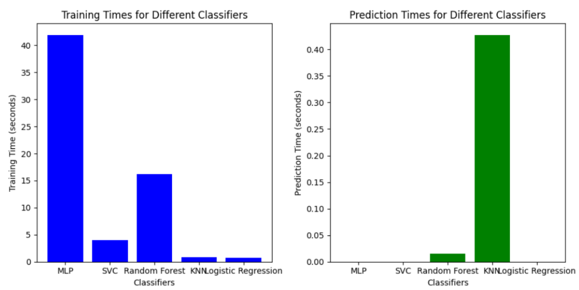
Optimization Results

Optimizing models and reducing their run times play a pivotal role in enhancing the efficiency and practicality of machine learning applications. The quest for model improvement is a continuous journey, and striking a balance between accuracy and computational speed is crucial.

Considering the MLP we employed, we found it had lengthy training times, but once trained, it exhibits remarkably swift prediction times. On the flip side, we encounter models like the K-Nearest Neighbors (KNN), renowned for its speedy training but notorious for high prediction times. This trade-off between training and prediction times underscores the importance of strategic optimization. The results of the various models' training and prediction times are further analyzed in Figure 6 below.

Figure 6

Training times of each model



Note. The above graphs represent the training and prediction times (in seconds) of each model used in our research.

In the realm of real-time applications or scenarios where responsiveness is paramount, minimizing prediction times is imperative. On the other hand, in cases where training time is a critical factor, such as continuous learning or updating models with new data, efforts should be directed towards optimizing the training phase.

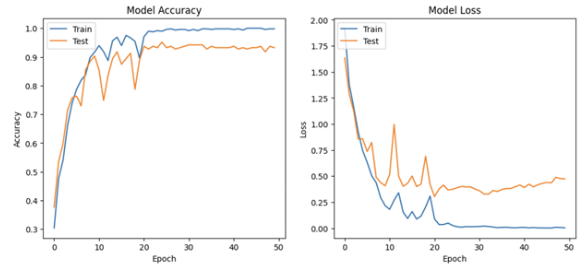
Strategic optimization not only enhances the overall performance of models but also makes them more adaptable to diverse use cases. It enables us to navigate the inherent trade-offs between training and prediction times, ensuring that machine learning solutions remain both accurate and efficient.

Overall Best Performance

The outcome for this facial recognition project stems from our expectation that CNN would emerge as the frontrunner in overall performance. Building on the success observed in previous projects, the CNN model stood out as a robust and effective choice for tackling facial recognition tasks. The decision to designate it as the anticipated leader was grounded in its proven track record of excelling in similar endeavors, making it a promising candidate for achieving high accuracy and reliability in the context of facial recognition. The results of this model are shown in more detail below (see Figure 7 and 8).

Figure 7

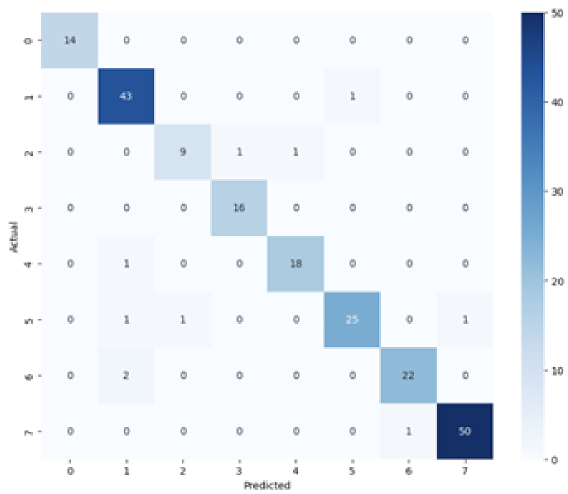
Convolutional Neural Network (CNN) model accuracy and loss



Note. The above figures depict the accuracy and loss for both the training and test data as the number of epochs increase.

Figure 8

Convolutional Neural Network (CNN) confusion matrix



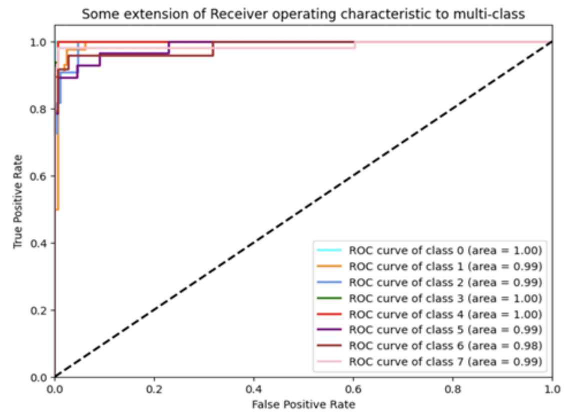
Note. The above image represents the confusion matrix from running the Convolutional Neural Network (CNN) on the eight different classes. The diagonal line shows the correctly predicted images per class.

Adding to our statement above that the CNN was our best performing model, please view the Receiver operating characteristic (ROC) curves and Precision-Recall curves shown below in Figures 9 and 10.

For both ROC and precision-recall curves, a good result is characterized by higher AUC values, a steep rise in the curve, and a favorable trade-off between sensitivity and specificity or precision and recall, depending on the context of the problem. The interpretation of goodness depends on the specific requirements of the machine learning task, and in this specific scenario the results were quite admirable.

Figure 9

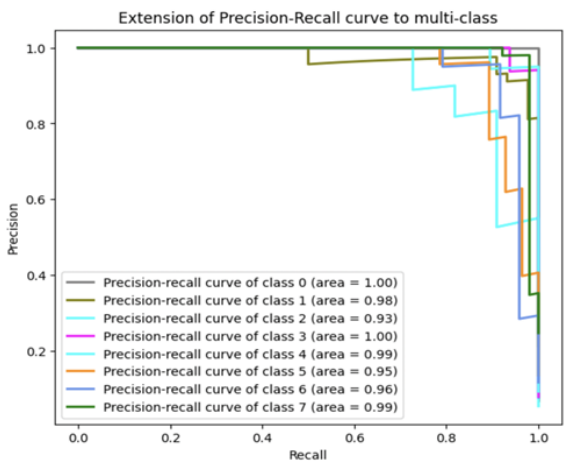
Convolutional Neural Network (CNN) ROC Curve



The above graph was used to measure the performance of the CNN model at all classification thresholds.

Figure 10

Convolutional Neural Network (CNN) Precision-Recall Curve



The above graph shows the tradeoff between precision and recall for different thresholds.

Conclusion

Comparison to Other Projects

We searched Kaggle for projects centered on face recognition, and what we found is that they mainly focused on emotion recognition across diverse faces. The majority of these projects employed CNN models, yielding commendable performance with accuracies consistently surpassing 90%. Given that our project involved the identification of distinct individuals' faces, it deviated slightly and demonstrated improved accuracy through the utilization of classification models.

While there were a couple of comparable projects for comparison, their methodologies varied, yielding superior results, reaching accuracies of 97% with specific models. Notably, these projects benefited from "cleaner" original images. Surprisingly, we couldn't locate another project that implemented RPCA to optimize models. Nevertheless, we are pleased to report that we achieved efficiency improvements without compromising accuracy.

Closing Thoughts

In our exploration of facial recognition system development, we used live face-captured images to enhance identity prediction accuracy. Employing various machine learning models and optimization techniques, including PCA and RPCA, optimized feature spaces and improved system performance. Our project provides insights into the effectiveness of machine learning algorithms and dimensionality reduction methods for creating robust facial recognition systems in security, access control, and human-computer interaction.

We carefully selected and implemented diverse algorithms, recognizing the unexpected performance of models like Random Forest before RPCA optimization. The crucial optimization phase significantly improved efficiency, balancing accuracy and computational speed through PCA and RPCA. The CNN emerged as the best-performing model, demonstrating robustness in facial recognition tasks, as confirmed by ROC and precision-recall curves.

Comparisons with other Kaggle projects highlighted our unique approach, emphasizing classification models and optimization techniques like RPCA. Despite some projects achieving higher accuracies with "cleaner" images, our focus on robustness and efficiency while maintaining accuracy sets our work apart.

In conclusion, this project reflects our commitment to advancing facial recognition systems. Through diverse techniques and meticulous model evaluation, we contribute to ongoing advancements, laying the groundwork for future real-world applications of facial recognition technology.

References (Project Comparisons)

Peldek, S. (n.d.). Face Recognition on Olivetti Dataset. Kaggle. https://www.kaggle.com/code/serkanpeldek/face-recognition-on-olivetti-dataset/input

lxyuan0420. (n.d.). Facial Expression Recognition using CNN. Kaggle. https://www.kaggle.com/code/lxyuan0420/facial-expression-recognition-using-cnn