Comparative Analysis of Machine Learning Models and Deep Learning Models in Glaucoma Detection

Dawson Hillebrand, Bibin John, Jonathon Machen

**Abstract**

*This study introduces a deep learning Convolutional Neural Network (CNN) method for early detection of glaucoma and compares its effectiveness with various non-CNN machine learning algorithms. Utilizing a comprehensive dataset of retinal scans, which includes 705 images of 300x300 pixels, the research explores several machine learning approaches. The study employs Support Vector Machine (SVM), Random Forest, and a selection of CNN architectures, all meticulously tested and validated against this dataset. Quantitative results are evaluated based on the specific performance metrics of each algorithm. Findings highlight the superiority of CNN models in glaucoma detection, achieving an impressive average accuracy of 93.49%. This outperforms the non-CNN algorithms, which averaged 83.5% accuracy across the board. The research demonstrates a significant improvement in detection accuracy with the CNN approach, reducing the likelihood of human error in diagnosing glaucoma. The results underscore the potential of CNNs in enhancing early glaucoma detection, which is crucial for preventing irreversible blindness.*

**1. Introduction**

Glaucoma, a leading cause of blindness worldwide, significantly affects the quality of life for those impacted. Characterized by damage to the optic nerve, often linked to increased intraocular pressure, this condition predominantly affects individuals over sixty, leading to irreversible vision loss if not detected and treated promptly. The importance of early detection is underscored by the irreversibility of optic nerve damage, necessitating proactive measures to preserve vision.

In the field of ophthalmology, detecting and diagnosing glaucoma presents a significant challenge. This challenge is heightened by the often-asymptomatic cases glaucoma, which render it difficult to diagnose without comprehensive eye exams. Consequently, there's a pressing need for innovative diagnostic approaches.

Machine learning and artificial intelligence are emerging as promising solutions in enhancing glaucoma detection and diagnosis. By employing machine learning technologies, notably Convolutional Neural Networks (CNNs), Support Vector Machine (SVM), and Random Forest algorithms, there's potential to substantially improve glaucoma detection's accuracy and efficiency. This improvement aids quick intervention and helps reduce clinical workload and potential human error in diagnoses.

Therefore, this project aims to assess the effectiveness of various machine learning models in early glaucoma detection. Utilizing a dataset of 705 retinal scans, the study implements and evaluates several machine learning strategies, including advanced deep learning CNN architectures and traditional algorithms like SVM and Random Forest. The research involves two phases: first, determining the accuracy of each model in glaucoma detection, and second, a comparative analysis to evaluate if CNN models surpass their non-CNN counterparts in this application.

The successful implementation of these models could yield multiple benefits. Medically, it could significantly reduce vision loss rates due to glaucoma. Economically, it might lower healthcare costs for late-stage glaucoma treatment. From a societal perspective, enhancing life quality for those at risk or afflicted by glaucoma is a noteworthy outcome. This project aspires to contribute to the medical field's ongoing efforts against glaucoma, leveraging advanced machine learning for societal welfare.

**2. Related Work**

The integration of machine learning in ophthalmology, especially in glaucoma detection, has garnered substantial research interest and provided valuable insights. A prominent example is Sejong Oh et al.'s work, detailed in "Explainable Machine Learning Model for Glaucoma Diagnosis." This study, achieving a significant 92.5% accuracy in glaucoma detection using machine learning, set a benchmark in the effectiveness of such approaches. Similarly, Marc Biarnes and his team, in "Classifying Glaucoma Exclusively with OCT: Comparison of Three Clustering Algorithms Derived from Machine Learning," reached an accuracy of 91.7%. These studies highlight machine learning's potential in glaucoma diagnosis, influencing the success benchmarks for this project.

While these studies lay the groundwork, this research diverges in its use of machine learning models, focusing specifically on CNN architectures. This study extensively explores various CNN architectures and optimizers, alongside traditional algorithms like SVM and Random Forest, offering a comparative approach not thoroughly pursued before.

A review of the literature also revealed potential issues in applying machine learning to medical diagnostics. High accuracy rates might indicate overfitting, a challenge this research actively avoids. By learning from these studies, best practices in model training and validation were implemented, ensuring robustness and reliability. This included comprehensive preprocessing, feature selection, and validation techniques to mitigate common problems like overfitting, aiming for models that are both accurate and practical in real-world scenarios.

This project builds upon existing knowledge and addresses a gap in comparative analysis between CNN and non-CNN models for glaucoma detection. Given glaucoma's complex and subtle symptoms, this exploration is vital, positioning machine learning as a potentially transformative tool in this field.

**3. Proposed Method**

In this study, three primary machine learning models were employed: a Convolutional Neural Network (CNN), a Support Vector Machine (SVM), and a Random Forest algorithm. These models were chosen for their proven capabilities in image classification tasks, which is essential for the analysis of retinal scans in glaucoma detection.

**3.1 Support Vector Machine (SVM)**

Support Vector Machines are supervised learning models that are effective in classification tasks. In our study, SVM is employed for binary classification: identifying whether a retinal scan indicates the presence of glaucoma or not. The key steps in this SVM implementation include:

* **Preprocessing**: Retinal images are converted to grayscale to reduce complexity and resized to a uniform size of 128x128 pixels, ensuring consistency in feature dimensions.
* **Feature Preparation**: Post resizing, images are flattened into 1D arrays suitable for SVM processing. Additionally, feature scaling is performed using the StandardScaler to standardize the data with zero mean and unit variance.
* **Model Training**: A linear kernel was used due to its efficiency and simplicity. The model is trained on the preprocessed dataset, optimizing for high accuracy while minimizing false negatives, critical in medical diagnosis.

**3.2 Random Forest**

The study incorporates Random Forest, an adaptable ensemble learning technique chosen for its robustness to overfitting, particularly in the domain of glaucoma detection. The methodology adopted in this research encompasses several fundamental stages:

* **Preprocessing**: The initial stage involves preprocessing the raw retinal images, which includes augmentation through rotation, width, and height shifts, and normalization through rescaling to 300x300 pixels to ensure standardized pixel values, thereby enhancing the model’s capacity for generalization.
* **Feature Extraction:** The images are transformed into flattened feature vectors to optimize the model’s capability to discern significant patterns and structures inherent in retinal scans.
* **Dimensionality Reduction:** Principal Component Analysis (PCA) efficiently manages computational complexities, reducing the feature space while retaining vital variance in the data.
* **Training and Optimization:** Through a GridSearchCV approach, the Random Forest classifier is trained on a preprocessed and dimensionality-reduced dataset. The model optimization process explores an array of hyperparameters, such as the number of estimators, tree depth, and bootstrap method, aiming to achieve a resilient balance between precision and generalization in glaucoma classification tasks.

**3.3 Convolutional Neural Network (CNN)**

This study further aimed to achieve the highest possible accuracy and the lowest possible recall for negative predictions. As such, seven distinct CNN pre-trained architectures, one hand-written “Vanilla” architecture, and six optimizers were chosen to ensure a wide range of models all working towards achieving the most ideal accuracies and recalls for the given dataset. The seven pre-trained CNN architectures were:

* **InceptionV3:** InceptionV3 was developed by Google and contains multiple parallel convolutional pathways as well as max pooling, which enables the design to capture features efficiently at different scales. These afford the architecture the unique capability to provide computational efficiency and improve the information flow by simultaneously capturing both local and global features.
* **VGG16 (Visual Geometry Group 16):** VGG16 is a deep convolutional neural network architecture designed by the Visual Geometry Group at the University of Oxford and consists of 16 layers, hence the name. VGG16 is known for its simplicity and uniform architecture, which allows for a unique combination of stability and flexibility, while also allowing for easy implementation.
* **VGG19 (Visual Geometry Group 19):** VGG19 is an extension of VGG16, with 19 layers. It shares a similar architecture but includes additional convolutional layers, making it deeper than its predecessor. While VGG19's increased depth allows it to capture more complex features, it also comes at the additional cost of higher computational requirements.
* **Xception:** Xception is another development from Google, as it is a direct extention of the InceptionV3 architecture. Differing from InceptionV3, Xception attempts to capture more complex patterns with fewer parameters and improving efficiency by replacing the standard convolutional with depth wise separable convolutions.
* **MobileNetV2:** The MobileNetV2 architecture is a pre-trained convolutional neural network that contains 53 layers. This architecture has been trained on over a million images and is quite adept at classifying images into a thousand categories, though it does require a image input size of 224x224, which forces the architecture to shrink the input images proved in this study.
* **NASNetLarge:** The Neural Architecture Search Network Large, or NASNetLarge, architecture is designed through reinforcement learning to and consists of a collection of cells, or repeated structures of normal and reductions connections, that allows for creations of complex and diverse architecture during the training process. In doing so, the NASNetLarge architecture provides a scalable and versatile architecture in the classification process.
* **DenseNet169:** The Densely Connected Convolutional Neural Network, or DenseNet, architecture is known for its connected blocks that allow for unique combinations of blocks that can scale the size of architectures. The DenseNet169 architecture is comprised of 169 layers. These layers make up varying types of blocks, including the ever-important bottleneck blocks made up of 1x1 and 3x3 convolutions. These allow for state-of-the-art performance when it comes to image classification and enables efficient information flow between the blocks and layers of the architecture.

Additionally, the six chosen optimizers were:

* **Stochastic Gradient Descent (SGD):** Stochastic Gradient Descent, or SGD, is a foundational optimization technique and will serve as a “baseline” to which all other optimizers will be compared to form the basis of this study. By its nature, SGD converges faster than traditional gradient descent due to its stochasticity, processing a smaller batch at each iteration as opposed to the entire dataset.
* **Adaptive Gradient Algorithm (Adagrad):** Adaptive Gradient Alrogthm, or Adagrad for short, is another optimizer used in machine and deep learning for training models and was designed to adjust the learning rates of each of the parameters, which has the bonus of being able to perform adequately well on sparse data, but the drawback of an increased potential of creating infinitesimal learning rates that decrease the convergence speed and can wreak havoc on the run times of the model.
* **Root Mean Squared Propagation (RMSProp):** Root Mean Square Propagation, or RMSProp, was developed to combat the limitations of the Adagrad optimization method. By ensuring that the changes to the learning rate are not as aggressive as the Adadelta learning rates could become. RMSProp works by giving more weight to recent gradients over past gradients, though this method is still not without its own limitations.
* **Adaptive Moment Estimation (Adam):** The Adaptive Moment Estimation, or Adam, optimizer is built off the SGD optimizer and was first introduced by Jimmy Ba and Diederik Kingma in “Adam: A Method for Stochastic Optimization,” a paper they wrote that was published in 2014. The Adam optimizer contains two exponential moving averages, one that is exponentially decaying average of past gradients and the other that is exponentially decaying average of past squared gradients. It also contains the concept of momentum, which accelerates the optimization process by adding part of the previous update to the current update. This was done to offset the limitations of the RMSProp optimization function.
* **Adadelta:** Adadelta was introduced in the paper “Adadelta: An Adaptive Learning Rate Method,” published in 2012 by Matthew Zeiler. Adadelta was conceived to adaptively adjust the learning rates during training based on historical gradient information. These rates are adjusted independently of each parameter, which allows for the model to converge more efficiently.
* **Nadam:** Nadam, short for Nesterov-accelerated Adaptive Gradient Descent, is an optimization technique that combines the Nesterov Accelerated Gradient and Adam optimization techniques into a single algorithm. This works by combining the Nesterov momentum into Adam’s learning rate approach. By combining these two techniques, the Nadam optimizer forms a more robust optimizer than either of the individual techniques on their own and holds the potential to yield accuracies of a must higher significance than the Adam optimizer technique on its own.

These forty-eight combinations of architecture and optimizer will allow for a complete overview of which model-optimizer combination was the best individually, as well as provide a method of comparing the architectures across all six of the optimizers, along with each optimizer across all eight architectures. By doing this, a combined system of analysis can be performed to determine the best possible choice when choosing the appropriate model when classifying retinal scans for the optimal accuracy and recall results.

**3.4 Experiments**

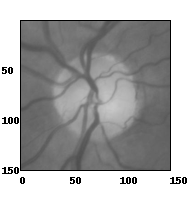
The experimental setup involves comprehensive data preparation and rigorous training and validation processes for each model. The dataset comprises high-resolution retinal scans, and techniques such as data augmentation were employed to ensure robust model training.

* **Dataset**: The study utilizes retinal scans, with each image being of 300x300 pixel resolution. The dataset includes both glaucoma and non-glaucoma cases, providing a diverse range for model training and validation.
* **Model Evaluation**: Performance metrics such as accuracy, precision, recall, and F1-score are used to evaluate each model. Special attention is given to the rate of false negatives, given the critical nature of glaucoma detection.

**4. Experiments**

**4.1 Dataset**

The dataset utilized in this study comprised 705 retinal scan images. Each image had a resolution of 300x300 pixels, providing high-quality data for accurate analysis. Of the 705 images, 396 are positive detection of glaucoma, and the other 309 are negative for glaucoma. The dataset was balanced with an appropriately equal number of glaucoma and non-glaucoma cases so as to prevent overfitting. Preprocessing steps for the SVM classifier included grayscale conversion to simplify the images, resizing to a uniform size for consistency, and standardization of pixel values to enhance model performance.



*Preprocessed Images*

**4.2 Support Vector Machine (SVM)**

In this study, the SVM model was trained on a subset of the retinal scan dataset, specifically 402 selected images. This subset was carefully chosen to represent a balanced mix of both glaucoma-positive and glaucoma-negative cases. The SVM implementation was conducted using Python’s sklearn library (as SVC, suitable for classification tasks). The experimentation approach included:

* **Kernel Functions**: Various kernel functions such as polynomial (poly), radial basis function (RBF), sigmoid, and linear were explored. This diversity in kernel functions allowed an assessment as to which best modeled the complex patterns in the dataset for glaucoma detection.
* **Gamma Parameter Tuning**: Different gamma values were experimented with. This parameter is critical in defining the influence of individual training examples on the decision boundary and overall model performance.
* **Regularization Parameter (C)**: Different C parameter values were tested. The C parameter helps in balancing the classification accuracy and the model's complexity, essential in preventing overfitting.
* **Standardization Methods**: Different standardization techniques were applied, including StandardScaler and MinMaxScaler, to normalize the data, ensuring no bias towards features of higher magnitude.
* **Feature Types**: Various feature types were tested, including intensity and density, shape, and texture using Local Binary Patterns (LBP). Trials with color images were also conducted to understand the role of color information in the classification accuracy.
* **Training and Evaluation**: The SVM model underwent rigorous training with these parameters and features on the chosen subset of 402 images. The model's performance was continually assessed using a validation set approach. Final evaluation metrics, such as accuracy\_score from sklearn.metrics, were employed to validate the effectiveness of the model for glaucoma detection.

**4.3 Random Forest**

Like the SVM model, the Random Forest model was trained using a 402-image subset of the original dataset with equal representation among both glaucoma positive and glaucoma negative cases.

PCA was employed as a means of transforming the high-dimensional image data into a lower-dimensional space while still retaining the information from the dataset. The model was built almost exclusively using sklearn’s in-built modules, specifically utilizing the RandomForestClassifier method using the default hyperparameters. These hyperparameters were then tuned using GridSearchCV, which is an exhaustive method to search through hyperparameter space in order to find the optimal configuration for the Random Forest classifier.

In this case, the parameter grid that was defined for the Random Forest included the number of trees, the max depth of each individual tree, the class weight, whether bootstrapping was used, the number of samples required to split an internal node, and finally the number of samples required to be at a leaf node. After running the cross-validation, the optimal set of hyperparameters was found to be:

* bootstrap: False – Indicating that the samples were not bootstrapped while building the trees.
* class\_weight: None – Indicating that no specific weight was used to define each class.
* max\_depth: None – Which implies that no maximum depth restriction was utilized for the decision trees.
* min\_samples\_leaf: 1 – Indicating that the minimum number of samples required to be at a leaf node was one.
* min\_samples\_split: 5 – Indicating that the minimum number of samples required to split an internal node was five samples.
* n\_estimators: 100 – Specifying that 100 decision trees were utilized.

Using this optimal set of hyperparameters, the Random Forest model was then evaluated by computing accuracy and then generating classification reports and a confusion matrix to visualize the precision, recall, F1-score, and support values for the model.

**4.4 Convolutional Neural Network**

Each model-optimizer combination was trained on the same data as both the SVM and Random Forest models, allowing for a more uniform approach across each of the three methods and ensuring that the arbitrariness of the random splitting of the training/validation/testing images was as minimal a factor in determination of the results as possible. On top of each of the pre-trained models, three additional layers were added on the top to account for the binary classification of determining the presence of glaucoma in each of the images. The images, in order, were:

* Flatten
* Dense layer of size 128 with a ReLu activation function
* Dense layer of size 1, with a Sigmoid activation function

To ensure as even a comparison amongst the architectures as could be had, the layers of each of the base models were frozen to prevent any additional training on the pre-trained architectures so that the results more accurately reflected the results of the architectures, and not the result of any additional training that would occur on the layers of the architectures.

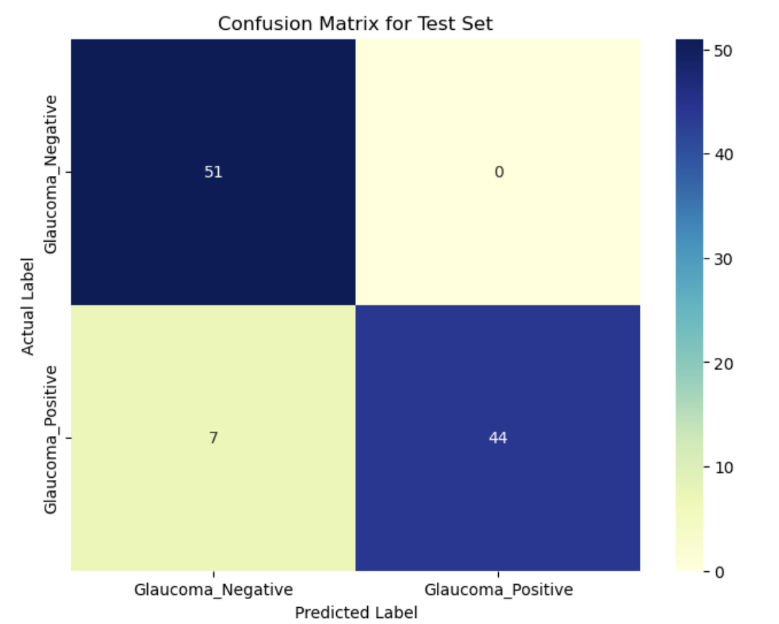
**5. Results**

**5.1 Support Vector Machine (SVM)**

The SVM model, crafted using SKLearn's SVC with a linear kernel and a default C parameter of 1, achieved an overall accuracy of 73.53% in glaucoma detection and an overall precision of 75%. This model was developed by preprocessing the images with StandardScaler, which normalized the pixel intensity and density, the primary features used for classification.

**5.2 Random Forest**

The Random Forest produced an accuracy score of 93.14%. In terms of precision, the model produced a 100% precision for non-glaucoma cases and an 88% precision for glaucoma cases. The recall for non-glaucoma was 86% while the recall for glaucoma cases was 100%. Overall, this suggests that the Random Forest model was fairly effective at predicting the presence of glaucoma from retinal scans, with a slight bias towards false negatives.



**5.3 Convolutional Neural Network (CNN)**

The CNN models, consisting of forty-eight combinations of architecture and optimizer, were, in the aggregate, 93.49% accurate with a precision for non-glaucoma cases of 96%. While the accuracy in the aggregate might not appear to be much more than a marginal improvement over the Random Forest classification model, the precision for non-glaucoma cases is a drastic improvement. This would suggest that on the whole, CNN models are better at ensuring a limited number of false negatives, though a rough overview doesn’t tell the full story. Further breaking down the accuracies for each model-optimizer combination and all non-glaucoma cases is required.

**Vanilla:**

* SGD
  + Accuracy: 89.22%
  + Precision for Non-Glaucoma Cases: 87.03%
* Adam
  + Accuracy: 98.04%
  + Precision for Non-Glaucoma Cases: 98.04%
* RMSProp
  + Accuracy: 94.12%
  + Precision for Non-Glaucoma Cases: 90.91%
* Adagrad
  + Accuracy: 85.29%
  + Precision for Non-Glaucoma Cases: 100%
* Adadelta
  + Accuracy: 81.37%
  + Precision for Non-Glaucoma Cases: 94.44%
* Nadam
  + Accuracy: 89.38%
  + Precision for Non-Glaucoma Cases: 80.95%
* Average
  + Accuracy: 89.38%
  + Precision for Non-Glaucoma Cases: 91.89%
* Standard Deviation
  + Accuracy: 5.47
  + Precision for Non-Glaucoma Cases: 6.51

These values would suggest that the “Vanilla” architecture was much less accurate of a model than the aggregate values would suggest. Additionally, while the precision for non-glaucoma cases was still lower than the aggregate values for the CNN models, it was still a better value than the Random Forest, which further hints at a step in the right direction as far as mitigating the risk of false negatives is concerned. Despite these improvements, the standard deviation of both the accuracy and non-glaucoma precision hints at a wide spread of values within the Vanilla architecture and optimizer combinations, spelling concern for repeatability in the models.

By far the best model in terms of accuracy for the Vanilla architecture was when used in tandem with the Adam optimizer. With an accuracy of 98.04%, the next best accuracy was four percentage points lower. But the optimizer is not without its imperfections. The precision for non-glaocuma cases was a tick or two lower than the best optimizer, though a precision of 98.04% might not appear to be a terrible result, there is a more precise combination in the Vanilla architecture when combined with the Adagrad optimizer, though the accuracy for this combination is far worse than the Vanilla-Adam combination, further implying that the Vanilla-Adam combination is the best combination within the scope of the Vanilla architecture.

**IncetpionV3:**

* SGD
  + Accuracy: 95.1%
  + Precision for Non-Glaucoma Cases: 94.23%
* Adam
  + Accuracy: 92.16%
  + Precision for Non-Glaucoma Cases: 92.16%
* RMSProp
  + Accuracy: 93.14%
  + Precision for Non-Glaucoma Cases: 95.83%
* Adagrad
  + Accuracy: 95.1%
  + Precision for Non-Glaucoma Cases: 97.92%
* Adadelta
  + Accuracy: 96.08%
  + Precision for Non-Glaucoma Cases: 97.96%
* Nadam
  + Accuracy: 96.08%
  + Precision for Non-Glaucoma Cases: 96.08%
* Average
  + Accuracy: 94.61%
  + Precision for Non-Glaucoma Cases: 95.69%
* Standard Deviation
  + Accuracy: 1.47
  + Precision for Non-Glaucoma Cases: 2.03

For the InceptionV3 architecture, the average accuracy improved in quite dramatic fashion compared to the Vanilla architecture, jumping an impressive five percentage points. This is further backed by the breakdown of each architecture-optimizer combination, as each accuracy was at or above 92%. Further evaluation shows that with a standard deviation at one and a half, the model is much more reliable in terms of getting a precision within a smaller range than the Vanilla architecture.

The best combination of architecture-optimizer for InceptionV3 was when it was paired with the Adadelta optimizer. Tied for the highest accuracy of the optimizers with the InceptionV3 architecture at 96.08%, the Adadelta outpaced the Nadam optimizer in terms of its added precision in detecting non-glaucoma cases at 97.96%, which is an improvement compared to the Nadam optimizer, which sits two percentage points behind at 96.08% precise. This proves that the best combination amongst the InceptionV3 architecture was when it was paired with the Adadelta optimizer

**VGG16:**

* SGD
  + Accuracy: 94.12%
  + Precision for Non-Glaucoma Cases: 95.92%
* Adam
  + Accuracy: 94.12%
  + Precision for Non-Glaucoma Cases: 92.17%
* RMSProp
  + Accuracy: 95.1%
  + Precision for Non-Glaucoma Cases: 95.83%
* Adagrad
  + Accuracy: 92.16%
  + Precision for Non-Glaucoma Cases: 97.78%
* Adadelta
  + Accuracy: 94.12%
  + Precision for Non-Glaucoma Cases: 94.12%
* Nadam
  + Accuracy: 88.24%
  + Precision for Non-Glaucoma Cases: 97.56%
* Average
  + Accuracy: 92.98%
  + Precision for Non-Glaucoma Cases: 95.62%
* Standard Deviation
  + Accuracy: 2.29
  + Precision for Non-Glaucoma Cases: 1.57

The VGG16 architecture took a small step back when compared to the InceptionV3 architecture, coming in at an overall accuracy amongst the combinations at 92.98%, approximately a point and a half behind the InceptionV3 architecture in terms of percentage points. It further still falls short of the Random Forest classification model, which spells a lack of hope in this architecture comparatively. Though the precision ofr non-glaucoma cases improved, the dip in accuracy means that this architecture would not be the one this study would suggest implementing into the field.

The most accurate combination of optimizer for the VGG16 architecture would be the RMSProp optimizer. With an accuracy of 95.1%, the RMSProp proved to be the best in terms of accuracy, but falls short upon further inspection of the precision, as a precision of 94.23% is still an improvement compared to the Random Forest classification model, the more precise architecture-optimizer combination falls to the Adagrad optimizer, which has an improved precision of 97.78%, but again falls short with an accuracy of 92.16%.

While still a step back in terms of accuracy compared to the InceptionV3 architecture, the VGG16 precision improved, though only just. The marginal improvement and accompanying marginal improvement in comparison to the standard deviation both provide hope that further architectures can improve upon the work that the previous architectures have laid out.

**VGG19:**

* SGD
  + Accuracy: 87.25%
  + Precision for Non-Glaucoma Cases: 79.69%
* Adam
  + Accuracy: 91.18%
  + Precision for Non-Glaucoma Cases: 86.21%
* RMSProp
  + Accuracy: 85.29%
  + Precision for Non-Glaucoma Cases: 78.13%
* Adagrad
  + Accuracy: 93.14%
  + Precision for Non-Glaucoma Cases: 92.31%
* Adadelta
  + Accuracy: 86.27%
  + Precision for Non-Glaucoma Cases: 78.46%
* Nadam
  + Accuracy: 86.27%
  + Precision for Non-Glaucoma Cases: 78.46%
* Average
  + Accuracy: 88.23%
  + Precision for Non-Glaucoma Cases: 82.21%
* Standard Deviation
  + Accuracy: 2.89
  + Precision for Non-Glaucoma Cases: 5.31

Oh, how the mighty have fallen. Talk about a major fall from grace. The VGG19 architecture plummeted face first off a cliff and hit every rock on the way down. A cataclysmic drop of seven percent compared to the already not spectacular Vanilla architecture in terms of its accuracy, the VGG19 architecture would have been a complete waste of time if not for the miraculous fourth quarter recovery of the Adagrad optimizer. Coming in at a lackluster 93.15% accurate with a corresponding 92.31% precision, this architecture-optimizer combination is at least marginally passable. But other than the Adagrad optimizer, the rest failed to make it to 90% in either accuracy or precision. Nothing further needs discussing, so moving on it shall be.

**Xception:**

* SGD
  + Accuracy: 96.08%
  + Precision for Non-Glaucoma Cases: 100%
* Adam
  + Accuracy: 97.06%
  + Precision for Non-Glaucoma Cases: 98%
* RMSProp
  + Accuracy: 96.08%
  + Precision for Non-Glaucoma Cases: 94.34%
* Adagrad
  + Accuracy: 98.04%
  + Precision for Non-Glaucoma Cases: 100%
* Adadelta
  + Accuracy: 97.06%
  + Precision for Non-Glaucoma Cases: 96.15%
* Nadam
  + Accuracy: 96.08%
  + Precision for Non-Glaucoma Cases: 97.96%
* Average
  + Accuracy: 96.73%
  + Precision for Non-Glaucoma Cases: 97.74%
* Standard Deviation
  + Accuracy: 0.73
  + Precision for Non-Glaucoma Cases: 2.02

Improvements. Improvements galore. With an average accuracy of 96.73% and an average precision of 97.74%, the Xception architecture is a drastic improvement from the VGG19 architecture and even far surpasses the previous best InceptionV3 architecture. With a standard deviation of almost half of the InceptionV3 architecture in terms of accuracy and a comparable standard deviation for the precision, the underlying values appear to suggest that the improvements are more than just surface level.

For the optimizers, the most accurate was the Adagrad optimizer, with an accuracy of 98.04%. It also had a corresponding precision of 100%, meaning that the model did not predict a single false negative. While the Stochastic Gradient Descent optimizer also had a precision of 100%, the accuracy was lower at 96.08%, and while those numbers are a far better improvement over even the InceptionV3 architecture, they fall short of the Adagrad optimization method.

**MobilNetV2:**

* SGD
  + Accuracy: 98.04%
  + Precision for Non-Glaucoma Cases: 98.04%
* Adam
  + Accuracy: 98.04%
  + Precision for Non-Glaucoma Cases: 100%
* RMSProp
  + Accuracy: 93.14%
  + Precision for Non-Glaucoma Cases: 100%
* Adagrad
  + Accuracy: 98.04%
  + Precision for Non-Glaucoma Cases: 100%
* Adadelta
  + Accuracy: 98.04%
  + Precision for Non-Glaucoma Cases: 100%
* Nadam
  + Accuracy: 97.06%
  + Precision for Non-Glaucoma Cases: 100%
* Average
  + Accuracy: 97.06%
  + Precision for Non-Glaucoma Cases: 99.67%
* Standard Deviation
  + Accuracy: 1.79
  + Precision for Non-Glaucoma Cases: 0.73

Overall, both the accuracy and the precision of the MobileNetV2 architecture show quality improvements over the Xception architecture. While the standar deviation of the accuracy for the MobileNetV2 architecture is higher than would be desired, nothing in the standard deviation would suggest that the accuracies achieved are anything other than significant improvements over the other architectures. The precision is also incredible, at 99.67% and suggests that the ability for the MobileNetV2 architecture to mitigate the risk of false negatives is unparalleled compared to the other architectures discussed.

The best architecture-optimizer combination is not as easy to determine when compared to the other architectures and optimizers. Within the MobileNetV2 architecture, there are three different optimizers that yield the same results in terms of accuracy and precision. The Adam, Adagrad, and Adadelta optimizers all yield accuracies of 98.04% and precisions of 100%. Further inspection of all three of the architecture-optimizer combinations shows that the predictions for each yield two false positives, zero false negatives, forty-nine true negatives, and fifty-one true positives. All three architecture-optimizer combinations are the same, though they have room for improvement. Further refinement of the top layers, or additions of more layers would be advised. Overall, the architecture proved to be exceptionally effective at classifying glaucoma cases from retinal scans.

**NASNetLarge:**

* SGD
  + Accuracy: 97.06%
  + Precision for Non-Glaucoma Cases: 98%
* Adam
  + Accuracy: 93.14%
  + Precision for Non-Glaucoma Cases: 98%
* RMSProp
  + Accuracy: 94.12%
  + Precision for Non-Glaucoma Cases: 94.34%
* Adagrad
  + Accuracy: 96.08%
  + Precision for Non-Glaucoma Cases: 100%
* Adadelta
  + Accuracy: 96.08%
  + Precision for Non-Glaucoma Cases: 96.15%
* Nadam
  + Accuracy: 93.14%
  + Precision for Non-Glaucoma Cases: 97.96%
* Average
  + Accuracy: 94.94%
  + Precision for Non-Glaucoma Cases: 97.74%
* Standard Deviation
  + Accuracy: 1.54
  + Precision for Non-Glaucoma Cases: 1.96

The NASNetLarge architecture proved to be quite effective, though not as effective as the MobileNetV2 architecture. With an average accuracy of 94.94% and average precision of 97.74%, the NASNetLarge architecture proved to be more effective than the Random Forest classification model.

The best architecture-optimizer combination for the NASNetLarge architecture is the Stochastic Gradient Descent optimizer, with both the highest accuracy at 97.06% and highest precision at 98%. While the precision of this particular combination is still lower than that of the “worst” performing architecture-optimizer combination within the MobileNetV2 architecture, it still outperforms the metrics of both the SVM and Random Forest models.

**DenseNet169:**

* SGD
  + Accuracy: 95.1%
  + Precision for Non-Glaucoma Cases: 92.59%
* Adam
  + Accuracy: 95.1%
  + Precision for Non-Glaucoma Cases: 92.59%
* RMSProp
  + Accuracy: 94.12%
  + Precision for Non-Glaucoma Cases: 97.87%
* Adagrad
  + Accuracy: 96.08%
  + Precision for Non-Glaucoma Cases: 94.34%
* Adadelta
  + Accuracy: 92.16%
  + Precision for Non-Glaucoma Cases: 100%
* Nadam
  + Accuracy: 91.18%
  + Precision for Non-Glaucoma Cases: 85%
* Average
  + Accuracy: 93.96%
  + Precision for Non-Glaucoma Cases: 96%
* Standard Deviation
  + Accuracy: 1.74
  + Precision for Non-Glaucoma Cases: 4.75

The DenseNet169 shows a step back from the NASNetLarge architecture and an even larger step back from the MobileNetV2 architecture, though both the average accuracy and precision are both higher than the Random Forest classifier at 93.96 and 96%, respectively. Overall, the DenseNet169 architecture further illustrates the improvements that the CNN architectures have over the machine learning Random Forest and SVM models.

There is not a clear choice when choosing the optimal combination of optimizer within the DenseNet169 architecture. The Adagrad optimizer achieves the highest accuracy at 96.08%, but the precision only managed 94.34%, whereas the Adadelta optimizer achieved a precision of 100%, but only managed a measly accuracy of 92.16%. This, coupled with the increased standard deviations between the accuracies and precisions suggests that the DenseNet169 architecture is less than reliable when compared to other CNN architectures.

**Optimizers:**

When comparing models, this study would be remise not to discuss the performance of each optimizer separate from the individual architectures.

* SGD:
  + Average Accuracy: 93.99%
  + Average Precision: 93.19%
* Adam:
  + Average Accuracy: 94.86%
  + Average Precision: 94.62%
* RMSProp:
  + Average Accuracy: 93.14%
  + Average Precision: 92.97%
* Adagrad:
  + Average Accuracy: 94.24%
  + Average Precision: 97.54%
* Adadelta:
  + Average Accuracy: 93.65%
  + Average Precision: 94.65%
* Nadam:
  + Average Accuracy: 92.04%
  + Average Precision: 91.73%

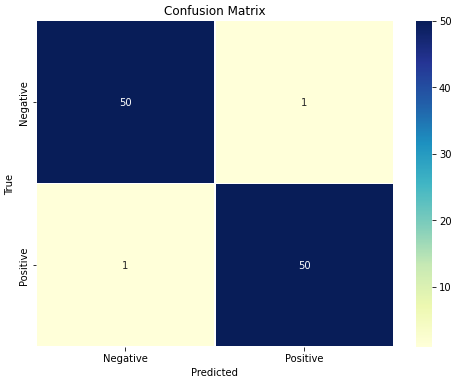
Overall, the optimizers all proved to be rather effective in the aggregate, with accuracies north of 92% across the board and precisions higher than 91%. When choosing the best optimizer, the clear winner would have to be the Adagrad optimizer. While the average accuracy is comparable to the Adam optimizer, the drastically increased precision that the Adagrad optimizer provides outweighs the slight decrease in accuracy.

**6. Conclusion**

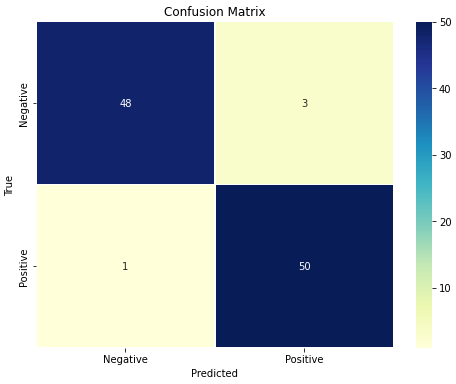
Overall, the Convolutional Neural Networks proved incredibly adept at classifying glaucoma cases within images of retinal scans passed through each architecture. With an overall accuracy of 93.49% and a precision of 94.12% across all forty-eight combinations, the CNN architectures proved more adept than both the SVM and Random Forest classifiers, which had accuracies of 73.53% and 93.14%, respectively and precisions of 75% and 88%, respectively. Comparitively, the CNN architectures vastly outperformed the Random Forest in terms of precision with comparative accuracies but outperformed the SVM in both accuracy and precision. But given the drastic differences in the combinations of architectures and optimizers, further inspection of the best combinations is required.

Below are the heatmaps of the confusion matrices for each of the best optimizer combinations with each architecture to provide a more granular conclusion as to the best architecture-optimizer combination.

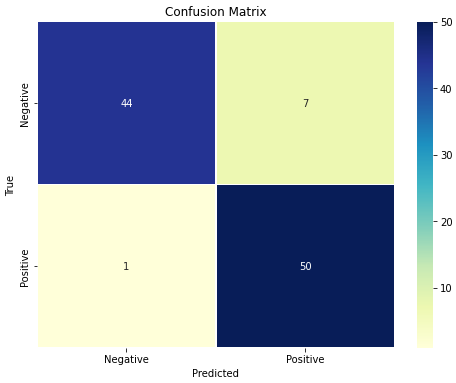
Vanilla architecture with Adam

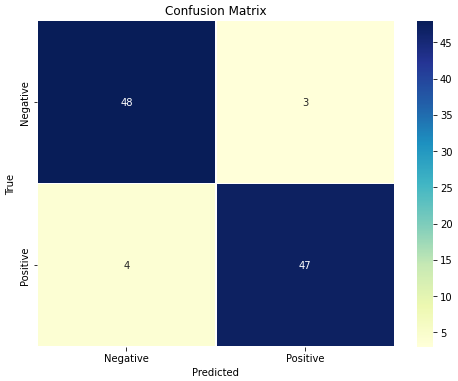


InceptionV3 architecture with Adadelta

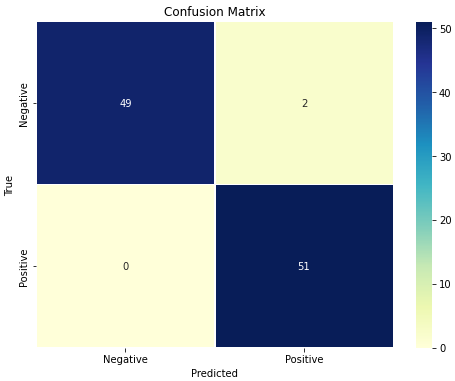


VGG16 architecture with Adagrad

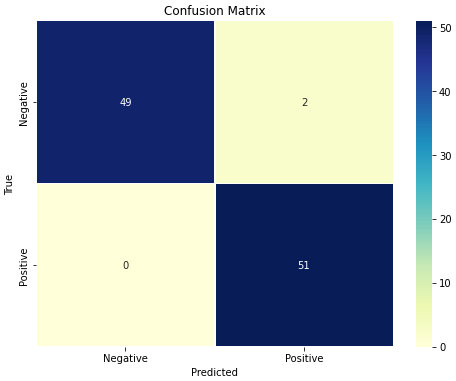
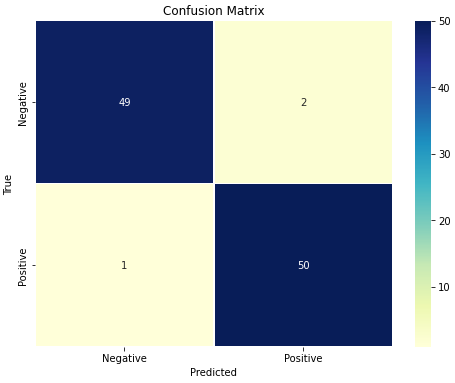


VGG19 architecture with Adagrad

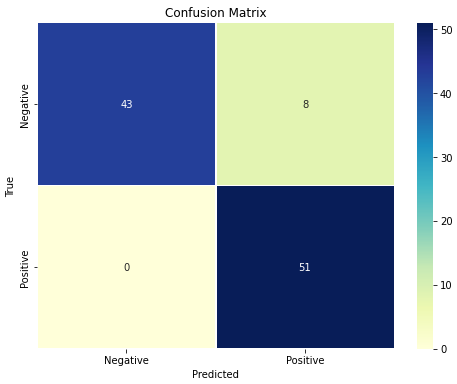
Xception architecture with Adagrad



MobileNetV2 with three optimizers

NASNetLarge architecture with SGD 

DenseNet169 with Adadelta



Based on the above confusion matrices, the two choices for the CNN architectures are either Xception paired with the Adam optimizer, or the MobileNetV2 architecture paired with the Adam, Adagrad, or Adadelta optimizers. All four combinations achieve the same accuracy and precision choices. These four proved to be the best combinations, but when taking a high-level view of the architectures the clear choice is the MobileNetV2 architecture, which outperformed the Xception architecture.

As such, comparing the MobileNetV2 architecture to the Random Forest and SVM models is the best course of action. With an overall accuracy of 97.06% and an overall precision of 99.67%, the MobileNetV2 architecture outperforms the Random Forest classifier by nearly four percent in terms accuracy and 11.67% in terms of precision. Compared to the SVM classifier, the MobileNetV2 architecture outperforms it by 23.53% and 24.67% in terms of accuracy and precision, repsectively. This not only suggests but proves beyond any doubt that the MobileNetV2 architecture is the far superior choice in terms of classification models when the architecture is paired with the Adam, Adadelta, and Adagrad optimizers. Couple the improved performance and the multiple choices in terms of optimizer, the MobileNetV2 architecture would be recommended for implementation in the field after further and rigorous testing and additional optimization occurs.

The findings emphasize the advantage of CNNs in diagnosing eye diseases from imaging data. Future research is directed towards applying these techniques to Optical Coherence Tomography (OCT) scans for a wider range of ocular conditions, potentially advancing diagnostic capabilities in ophthalmology significantly.

**Contributions**

The SVM models and results were compiled by Dawson Hillebrand.

The Random Forest models and results were compiled by Bibin John.

The CNN models and results were compiled by Jonathon Machen.

The project report was a collaboration between the three.

**References**

[1] Biarnés, M., Ventura-Abreu, N., Rodríguez-Una, I. et al. Classifying glaucoma exclusively with OCT: comparison of three clustering algorithms derived from machine learning. *Eye* (2023). <https://doi.org/10.1038/s41433-023-02785-5>

[2] Oh, S., Park, Y., Cho, K. J., & Kim, S. J. (2021). Explainable Machine Learning Model for Glaucoma Diagnosis and Its Interpretation. Diagnostics (*Basel, Switzerland*), 11(3), 510. <https://doi.org/10.3390/diagnostics11030510>

[3] Kingma, Diederik P. and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” *CoRR* abs/1412.6980 (2014).

[4] Zeiler, Matthew D.. “ADADELTA: An Adaptive Learning Rate Method.” *ArXiv* abs/1212.5701 (2012).