Eye Disease Classification

Tyler Travis
Software Engineering
3643127
ttravis@unb.ca

Cooper Dickson
Software Engineering
3684104
cdickson@unb.ca

***Abstract*—The Eye Disease Classification project employs machine learning techniques, specifically Convolutional Neural Networks (CNNs), for enhanced diagnostic capabilities in diverse eye conditions. Informed by insights from the literature review emphasizing the effectiveness of CNN models in image classification, the project utilizes a carefully curated dataset featuring Normal, Diabetic Retinopathy, Cataract, and Glaucoma retinal images. Over the course of the study, three distinct CNN architectures undergo scrutiny, with variations in hyperparameters (16, 32, 64) and (16, 32, 64, 128) and adjustments to kernel size. The model achieves a noteworthy accuracy of 85% with a corresponding loss function of 0.25, demonstrating its efficacy in classifying various eye conditions. The research serves as a practical exploration at the intersection of machine learning and biomedical retinal research, contributing valuable insights to both domains.**

I. Project Introduction

Eye-related ailments, such as Diabetic Retinopathy, Cataracts, and Glaucoma affect over 2.2 billion people [1] across the globe. As imaging is an important component of assessment, machine learning has become an increasingly powerful decision support tool that the Eye Disease Classification project will leverage. The endeavor to combat and understand various eye diseases holds strong significance, in pursuit of improving the quality of life for many people with the quintessential sense of eyesight.

The cornerstone of this project lies in the utilization of the Eye Disease Retinal Images Data set. It includes retinal images categorically representing Normal, Diabetic Retinopathy, Cataract, and Glaucoma conditions, each comprising approximately 1000 samples that are labeled in Figure 1. These images are sourced from a diverse array of esteemed repositories such as IDRiD, Ocular Recognition, and HRF, thus ensuring a rich and diverse representation of eye diseases. Figure 1 denotes an example of each eye classification with its corresponding label.

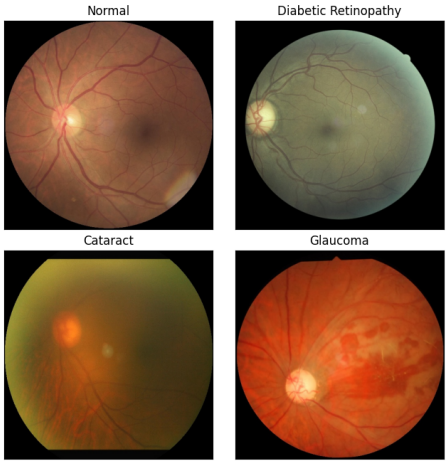


Fig. 1: **Retinal Conditions in the Left Eye** This composite grid showcases four distinct retinal conditions The top-left quadrant features a healthy left eye retina representing the ”Normal” class, while the top-right quadrant illustrates Diabetic Retinopathy, the bottom-left quadrant showcases a cataract, and the bottom-right quadrant displays an example of glaucoma.

The motivation behind working with this dataset stems from the formidable challenges associated with acquiring comprehensive biomedical data on eye diseases, given ethical and practical constraints related to human subjects. The utilization of this existing dataset provides a unique opportunity to make meaningful contributions to the field. Beyond the immediate application to eye diseases, developing a classifier for this dataset serves as a foundational step in deepening our understanding of essential concepts in modern deep learning and neural networks. This undertaking is not only about addressing a specific problem but also serves as a robust learning experience, allowing our team to navigate the intersection of machine learning and biomedical retinal research. This introduction aims to succinctly convey the rationale for our work and its broader implications.

II. Literature Review

A. Gap Statement

While substantial progress has been made in the field of eye disease classification, leveraging machine learning techniques, a significant gap persists in the current state of research. Existing studies often focus on specific eye diseases in isolation, utilizing relatively small data sets with limited diversity. The data set at hand, comprising Normal, Diabetic Retinopathy, Cataract, and Glaucoma retinal images, each with approximately 1000 images collected from diverse sources like IDRiD, Oculur recognition, and HRF, presents a valuable opportunity to address this gap. Despite its richness and diversity, there remains a dearth of comprehensive research that exploits the full potential of this data set to develop unified and robust machine learning models capable of not only accurate disease diagnosis but also nuanced classification among these various eye conditions.

B. Modern Common Practice

Understanding the foundational processes involved in diagnosing patients with eye diseases is of utmost importance in the realm of optometry. It is essential to understand the various strategies for identifying such eye diseases. Understanding these alternatives will help interested parties understand if this provides fundamental value to the health care system. Optometrists typically commence their diagnostic journey with a visual examination, where they evaluate the patient's vision while observing the eye for any discernible abnormalities [2]. Another crucial diagnostic tool in their arsenal is slit-lamp biomicroscopy, a method specifically designed for investigating the anterior segment of the eye. This examination proves invaluable in identifying a spectrum of issues, often pinpointing irregularities in the health, thickness, or surface of the cornea, thus contributing to disease diagnosis [3]. Furthermore, Optometry is an essential assessment that quantifies intraocular pressure (IOP), playing a pivotal role in screening for conditions such as glaucoma and retinopathy [4]. Ophthalmology, another vital diagnostic procedure, offers a comprehensive view of the retina, optic disc, and blood vessels [5]. Its efficacy in identifying various eye diseases, coupled with its aptitude for pattern recognition, positions it as a valuable diagnostic tool. Notably, this parallels the eye disease retinal image data set utilized in this project, underscoring the relevance and significance of these diagnostic practices in contemporary optometry.

C. Machine Learning Techniques

To ensure the best results when predicting eye disease, the latest machine learning techniques, like deep learning, are being used. A popular model used to classify images is the Convolutional Neural Network(CNN) model, which is a neural network that specializes in processing data in a grid like format, which is used for identifying objects and patterns in visual data [6]. To use this model a user would need to give the CNN an image, and it gives a value to the corresponding subset based on its color density [7]. The CNN model must stack multiple convolution layers with various filter sizes and pixels for kernel size, along with an activation function, rectified linear unit function (ReLu), to evaluate color patterns within images of an iris [8]. ReLu is used to interpret the positive part of its argument, which is why it is the most popular activation function [9]. Another method is to evaluate Optical coherence tomography (OCT) images, which are cross section images of a retina, to find irregular shapes that are associated with different eye diseases [10]. sing A Deep Learning CNN model's reliability can be integrated with Computer-Aided Diagnosis (CAD) for the detection and classification of Glaucoma, Diabetic Retinopathy, and Age-Related Macular Degeneration. [11]. Research indicates that the CNN models are the most popular for image classification thus is the leading architecture for work related to this project.

Although CNN models are favoured for this type of work, different techniques can be used to classify eye disease images. The use of thermal imaging of eyes can be leveraged to detect diabetic eye disease. Thermal images can be converted from RGB to Gray to be used in a feature extraction model to extract features like texture and color moments, which are used in K-Fold Validation technique to be classified [12]. Diseases could also be predicted based on the Optic Disk(OD) localization in the retina, where blood vasculatures and nerve axons connect to the eye. A CNN two-stage training technique is applied to this specific project to classify the OD localization [13]. Another approach to this work was to use color fundal images as inputs to find vascular and nonvascular abnormalities which was classified into one of 15 different disease classes [14]. In order to classify these images, different algorithms could be applied, where in one study the results were compared between Support Vector Machine (SVM), Random Forest, Bagging, and Gradient Boosting Decision Tree. In this study the SVM and Random Forest yielded the highest results [15].

Traditional machine learning algorithms can be just as effective as deep learning models when combined with crafted practical feature selection. Features like age, cornea size, and discoloration are decisive factors when classifying eye disease [16]. Textual data of the eye is another feature that proved to be applicable when classifying images, because it provides an alternative view when patterns within a retina are difficult to gauge or if the images vary in visibility [17]. Although there are different factors that impact the likeliness of eye disease, three factors that are applicable to this project are color fundus, vascular patterns, and measurement of the optic disk. These features were used with the algorithms Naive Bayes and SVM along with Cohen's Kappa of 10-fold cross-validation were used when comparing glaucoma screening results with their own classifier [18]. The selection of features is a vital importance when training a machine learning classifier for medical purposes, as it ensures the accuracy and reliability is performing at the highest standard.

III. Methods

IV. Model Architecture Definition

After reading most research suggesting convolutional neural networks [6], Three distinctive Convolutional Neural Network (CNN) architectures, denoted as *CNN A*, *CNN B*, and *CNN C*, have been meticulously crafted to extract intricate features from the eye-related image dataset. These architectures follow a sequential structure, defined using the Keras Sequential model.

For *CNN A*, the architecture commences with a convolutional layer utilizing 16 filters of size (3, 3) and a rectified linear unit (ReLU) [9] activation function. Subsequently, max-pooling layers with a pool size of (2, 2) are applied. This pattern repeats for two additional convolutional layers with 32 and 64 filters, respectively. Spatial dimensions are systematically reduced through max-pooling after each convolution [19]. The flattened representation is then connected to a dense layer of 128 units activated by ReLU, culminating in an output layer with units equal to the number of classes, activated by the softmax function.

In the case of *CNN B*, the architecture closely mirrors *CNN A* but introduces an additional convolutional layer with 128 filters. This augmentation aims to enhance the network's capacity to capture intricate features. The subsequent layers maintain the structure of *CNN A*, concluding with a softmax-activated output layer.

CNN C, distinguished by a different filter size (5, 5) in the initial convolutional layer, seeks to capture broader spatial features. The subsequent layers, including max-pooling and convolution, continue in a manner similar to the previous architectures. The final dense layers and output layer maintain consistency with the ReLU [9] and softmax activations [20], respectively.

These architectures, with their varying filter sizes and depths, aim to provide a nuanced exploration of the dataset. The models are designed to learn and distinguish features associated with different eye conditions, and subsequent training and evaluation phases seek to shed light on the effectiveness of these architectural choices.

V. Model Compilation and Training

The models are meticulously compiled using the Adam optimizer and categorical crossentropy loss. Subsequently, each model undergoes an intricate training process spanning 10 epochs, with validation conducted on a distinct set (`validation_data=(test_images, test_labels)`). Matplotlib is employed for visualization, tracking and plotting the training history, including accuracy and loss metrics over epochs.

VI. Model Evaluation and Metrics Calculation

Post-training, the models are rigorously evaluated on the test set. This evaluation extends beyond basic accuracy, encompassing the calculation of essential metrics such as sensitivity, specificity, precision, NPV (Negative Predictive Value), and F1 score. These metrics provide a nuanced understanding of the models' performance on each class, offering valuable insights into their predictive capabilities [21].

VII. Avoiding Overfitting

To gain insights into the model's learning process and potential overfitting, we systematically plotted the difference between training and validation accuracy across multiple training epochs. This analytical approach allowed us to visualize the divergence or convergence of performance metrics [22], providing valuable information on the model's ability to generalize to unseen data and identifying optimal training epochs for robust model performance.

VIII. Confusion Matrices and Visualization

To facilitate interpretability, the code generates confusion matrices, visually representing the models' predictions across different classes. This visual aid serves as a powerful tool for discerning patterns of misclassifications and gaining a deeper understanding of the models' behavior.

IX. Model Comparison

A comparative analysis is a central aspect of the code, wherein three distinct CNN architectures undergo training and evaluation. This comparative approach allows for a detailed examination of how architectural choices impact model performance, shedding light on which configurations prove more effective for the specific characteristics of the eye-related image dataset.

X. Library Usage and Additional Tools

TensorFlow and Keras form the backbone of model construction and training, while Matplotlib facilitates the visualization of training history. Scikit-learn is leveraged for metrics calculation and data preprocessing, adding a layer of robustness to the overall workflow. The combination of these tools and methodologies creates a thorough, well-structured approach to CNN modeling and evaluation.

XI. Results

A. Accuracy by Class

In this section, we present a detailed analysis of the model's performance across different classes, specifically focusing on its accuracy in classifying images related to Cataract, Diabetic Retinopathy, Glaucoma, and Normal cases. The visual representation in Figure 2 provides a comprehensive overview of the classification accuracy's achieved by our model across various convolutional neural network (CNN) architectures. These CNN architectures differ in terms of layer depth, layer sizes, kernel size, activation functions, and number of epochs which influence their ability to capture different features in

the eye disease images. The figure is thoughtfully crafted to convey essential information about the model's discriminative capabilities for each class, allowing for a quick comparison of performance across different conditions. These results are used to make informative decisions on which model is optimal at predicting specific classes as well as the overall accuracy.

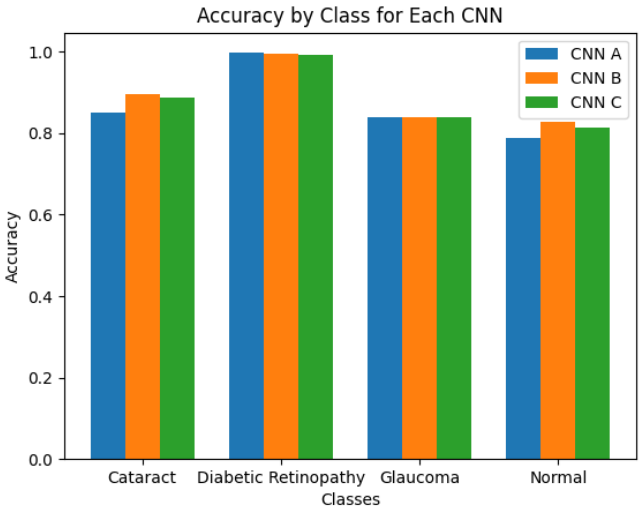


Fig. 2: **Accuracy By Class**: Cataract, Diabetic Retinopathy, Glaucoma, and Normal classification accuracy's described over different CNN architectures.

B. Training Dynamics

In the exploration of the model's performance, we delve into the training dynamics across various architectures, shedding light on the evolution of training accuracy over successive epochs. Figure 3 serves as a visual representation, allowing for a comparative analysis of how the training accuracy improve throughout the training process. The graph encapsulates the overarching trends observed across different convolutional neural network (CNN) architectures, providing a overview of their learning dynamics. This approach aims to convey a comprehensive understanding of how each architecture evolves in terms of training accuracy over the course of training epochs.

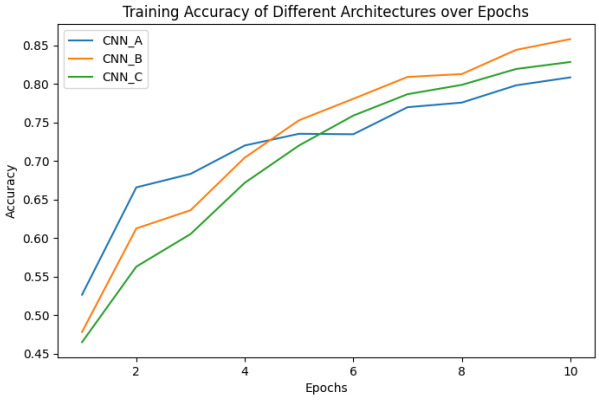


Fig. 3: **Training Accuracies over Epochs**: All architectures comparatively, as trained over epochs, seeing how validation accuracy improves

C. Training Loss Comparison

In this segment, we embark on an insightful exploration of the training dynamics by examining the evolution of the training loss function across various convolutional neural network (CNN) architectures. Figure 4 serves as a visual compass, offering a comparative analysis of the training loss over successive epochs. The graph encapsulates the collective trajectory of different architectures, providing a concise overview of how the loss function decreases throughout the training process.

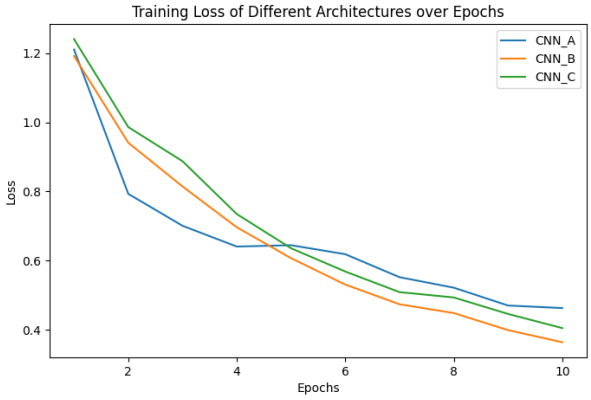


Fig. 4: **Training Loss Function over Epochs**: All architectures comparatively, as trained over epochs, seeing how loss function decreases.

D. Metrics by Class

1) Sensitivity: Figure 5 serves as a visual representation, offering insights into the sensitivity of three different convolutional neural network (CNN) architectures concerning varying classes. The graph delves into the nuanced aspect of how well these models accurately detect the ratio of true positives to the sum of true positives [23], thereby providing a measure of

their sensitivity. Strong sensitivity implies the model's ability to effectively capture true positive cases within each class. As we navigate through the ensuing discussion, the emphasis will be on elucidating unique observations revealed by the metrics across classes, accompanied by concise comments to highlight key findings. Through this exploration, our goal is to offer a nuanced understanding of how each CNN architecture performs in terms of sensitivity across different classes, contributing to a holistic appraisal of their classification capabilities.

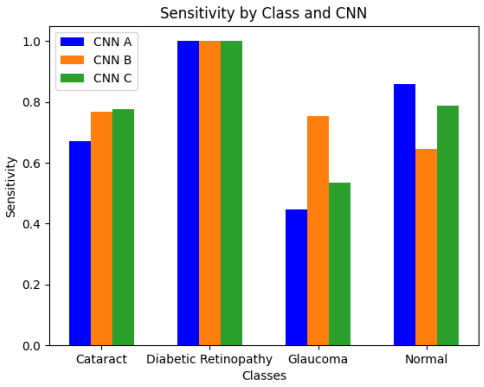


Fig. 5: **Sensitivity with Changing Class and CNN Architecture**: This observation suggests how well three models can accurately detect the ratio of true positives to the sum of true positives. A strong sensitivity suggests the model can accurately capture true positive cases effectively.

2) Specificity: In this analysis of model performance, we shift our focus to a critical metric—specificity—examining the model's ability to accurately identify negative instances. Figure 6 provides a visually insightful representation, illustrating how specificity varies across different classes and convolutional neural network (CNN) architectures. The graph details the model's effectiveness in correctly identifying true negatives, expressed as the ratio of true negatives to the sum of true negatives [23]. A robust specificity signifies the model's skill in avoiding false positives within each class. Throughout our subsequent discussion, we aim to uncover distinctive insights revealed by specificity metrics across classes, accompanied by concise comments to highlight key observations. This exploration seeks to offer a nuanced perspective on how each CNN architecture performs in terms of specificity across a range of classes, contributing valuable insights to the comprehensive evaluation of their classification capabilities.

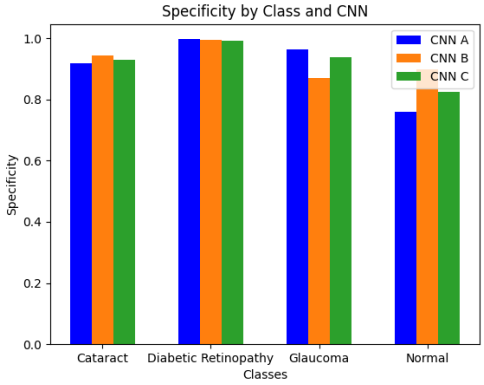


Fig. 6: **Specificity with Class and Architecture**: This observation suggests how well the classification model can correctly identify negative instances divided by the sum of true negatives. A strong specificity describes a model effectively avoiding false positives.

3) Precision: In examining model performance, our attention turns to the metric of precision, specifically delving into the accuracy of positive predictions made by a classification model. Figure 7 provides a visual representation, illustrating the relationship between precision, different classes, and convolutional neural network (CNN) architectures. The graph showcases the model's precision by gauging its ability to make accurate positive predictions, where a higher precision result corresponds to a lower false positive rate. As we proceed with the discussion, our goal is to uncover insights revealed by precision metrics across various classes, offering brief comments to highlight key observations. This exploration contributes to a nuanced understanding of how each CNN architecture performs in terms of precision across different classes, adding valuable insights to the overall evaluation of their classification capabilities.

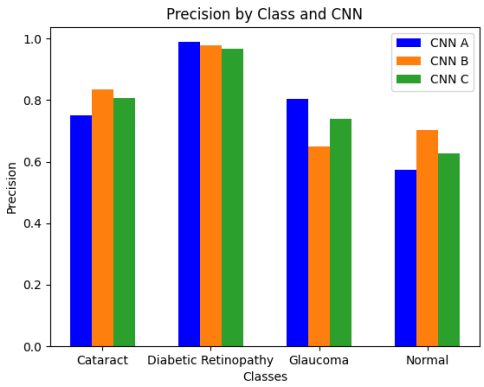


Fig. 7: **Precision with Class and Architecture**: The accuracy of positive predictions made by a classification model, which in terms of a high result will yield a low false positive rate.

4) F1 Score: In this examination of model performance, we turn our focus to the critical metric of F1 Score, providing a comprehensive representation of both precision and sensitivity within the context of different classes and convolutional neural network (CNN) architectures. Figure 8 serves as an illustrative visual guide, portraying the nuanced interplay between F1 Score, class distinctions, and architectural variations. The graph encapsulates the balance between precision and recall for each class, with an ideal scenario characterized by a high F1 Score indicating a harmonious equilibrium between these two metrics. As we delve into the subsequent discussion, our aim is to unravel insights offered by F1 Score metrics across diverse classes, complemented by comments to highlight key observations.

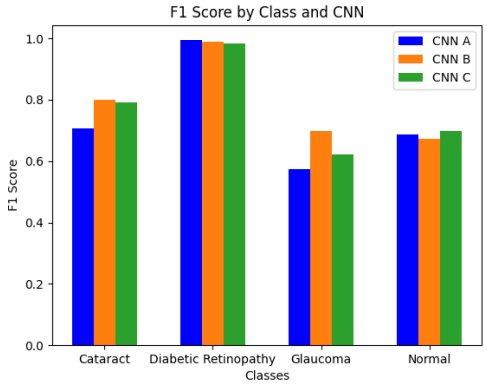


Fig. 8: **F1 with Class and Architecture**: A representation of both the precision and sensitivity, which varies from class to class. Ideally there is a good balance between precision and recall, which is represented by a high $F1$ score.

5) Negative Prediction Value: In this assessment of model performance, our focus centers on the metric of Negative Predictive Value (NPV), examining its variation across different classes and convolutional neural network (CNN) architectures. Figure 9 serves as a visual representation, offering insights into the NPV dynamics within the context of class distinctions and architectural nuances. The graph provides a snapshot of how well the model accurately predicts negative instances, emphasizing the interplay between NPV, class-specific characteristics, and CNN architecture. As we navigate through the ensuing discussion, our goal is to unravel unique insights revealed by NPV metrics across diverse classes. Though currently without additional details, our exploration aims to contribute to a nuanced understanding of how each CNN architecture performs in terms of Negative Predictive Value across various classes.

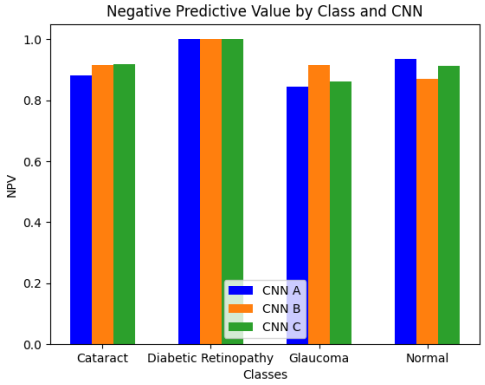


Fig. 9: **NPV with Class and Architecture**: A representation of the proportion of true negative test results compared to all negative results.

E. Optimal Epochs

Finding the optimal number of epochs while training a CNN is a vital step to training efficient models and to achieve optimal validation results as well.

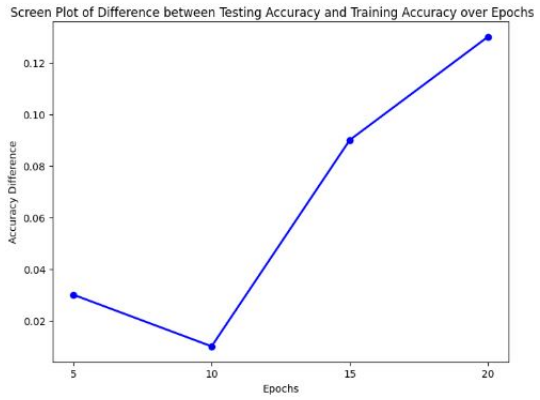


Fig. 10: **Plot of Difference between Training and Validation Accuracy over Multiple Epoch Training Sizes.**

F. Confusion Matrix

The primary purpose of employing a confusion matrix Figure 11 in our analysis is to provide a detailed and nuanced evaluation of the performance of our convolutional neural network (CNN) in image classification. This was done with the best performing model (*CNN B*) Unlike simple accuracy metrics, a confusion matrix breaks down the model's predictions into specific categories, such as cataracts, diabetic retinopathy, glaucoma, and normal eyes. By comparing the true labels with the predicted labels, the confusion matrix enables us to identify not only the overall accuracy of the model but also the specific areas where it excels or struggles. This level of granularity is crucial for understanding the model's strengths and weaknesses in differentiating between distinct pathology classes.

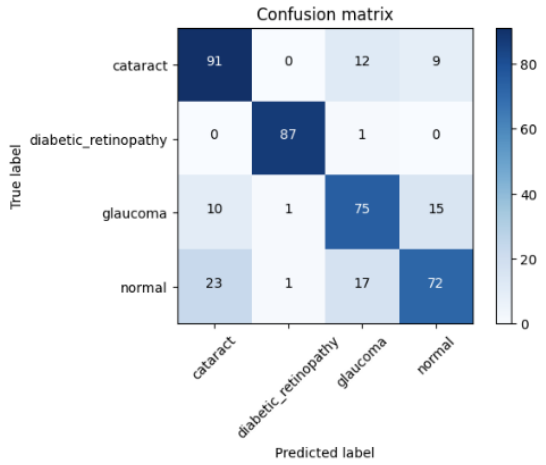


Fig. 11: **Confusion Matrix**: A two dimensional matrix representing the true label compared to the predicted label, of cataracts, diabetic retinopathy, glaucoma, and normal eye images.

XII. Discussion

A. Accuracy by Class Observations

Diabetic Retinopathy consistently performs exceptionally well across all three CNN architectures, nearly achieving perfect accuracy. This suggests that the features crucial for identifying Diabetic Retinopathy are effectively captured by the designed architectures (Figure 2). The minor variations between *CNN A, B, and C* may be attributed to the nuanced differences in their structures. Normal classification, with an average accuracy around 80%, indicates a more challenging task for the CNNs. The team is looking into continuing this project, and looking at ways to pre-process the normal features in a more meaningful way to avoid mixing up with other classifications [24]. The variation in performance, where *CNN B* excels and *CNN A* performs slightly worse, suggests that the additional convolutional layer in *CNN B* contributes positively to distinguishing normal cases. The 80% accuracy highlights the complexity of identifying normal eye images, potentially due to the diversity within the normal class. Uniformly high accuracy across all three CNNs for Glaucoma indicates a robust performance in detecting this condition. The consistent measurement suggests that the chosen architectures effectively capture features relevant to Glaucoma. The absence of significant variations between CNNs implies that the complexity of Glaucoma detection is equally well-handled by all three architectures.

B. Training Dynamics

Analyzing the training accuracy trends across epochs for *CNN A*, *CNN B*, and *CNN C* reveals distinctive patterns (Figure 3). *CNN A* starts with a strong performance but experiences diminishing growth in accuracy over time as it asymptotically plateaus, suggesting potential over-fitting or convergence to sub-optimal solutions. The implementation of early stopping mechanisms [25] could mitigate over-training issues and enhance generalization. *CNN B* demonstrates late-blooming strength, indicating the potential benefits of fine-tuning and longer training epochs [26]. Further exploration of optimal hyper-parameters may contribute to improved performance. *CNN C*, despite a slow start, exhibits adaptability. Understanding the specific architectural elements influencing each CNN's performance suggests the importance of architectural refinement.

C. Training Loss Comparison

The performance differences among *CNN A*, *CNN B*, and *CNN C* are shaped by their architectural designs (Figure 4). *CNN A* initially shows strong feature extraction capabilities but suffers from increased loss, indicating potential overfitting or convergence issues [27]. *CNN B*, with an extra convolutional layer, strikes a balance between complexity and generalization, making it suitable for fine-tuning and extended training. *CNN C*, featuring larger filters, adapts to dataset complexity, starting with high loss but transitioning to a mid-range position. These findings highlight the crucial role of architectural choices in influencing the learning dynamics of convolutional neural networks.

D. Metrics by Class

1) Sensitivity: The distinct sensitivity patterns across Cataract, Diabetic Retinopathy, Glaucoma, and Normal classes reveal notable variations among *CNN A*, *CNN B*, and *CNN C* (Figure 5). For Cataract, CNN B demonstrates the highest sensitivity, outperforming both *CNN A* and *CNN C*. In Diabetic Retinopathy, all three CNNs exhibit strong and uniform sensitivity, suggesting robust performance in identifying positive instances for this class. The sensitivity analysis becomes particularly intriguing in Glaucoma detection, where *CNN B* significantly outshines *CNN A* and *CNN C*. *CNN B* achieves nearly 80% sensitivity, while *CNN C* and *CNN A* lag behind at 50% and 40%, respectively. This substantial difference underscores *CNN B*'s superior ability to accurately identify positive instances for Glaucoma, hinting at its effective feature extraction and architectural suitability for this condition [28]. The performance in the Normal class follows a different trend, with CNN B displaying the lowest sensitivity compared to *CNN A* and *CNN C*, both reaching around 80%.

2) Specificity: The specificity results emphasize the crucial role of architectural choices in shaping the models' ability to accurately identify negative instances across various eye disease classes. The consistent and high specificity for Cataract suggests a robust capacity across all CNNs to correctly discern true negative cases (Figure 6). The exceptional performance in specificity for Retinopathy, although slightly varied, underscores the reliability of all three architectures in avoiding false positives. However, the noteworthy difference in Glaucoma, with *CNN B* trailing behind, signals potential challenges in correctly identifying negative instances for this condition. This variation may be attributed to architectural nuances impacting feature recognition. In the case of Normal images, the superior

specificity of *CNN B*, followed by *CNN C* and *CNN A*, highlights the positive impact of additional convolutional layers on enhancing the models' discriminatory capabilities.

3) Precision: The consistently high precision for cataract across all CNNs suggests that the models excel in accurately identifying this particular eye condition without generating false positives. This robust performance may indicate that the features associated with cataract are distinct and easily discernible, allowing the models to make precise predictions (Figure 7). However, the variation in precision across CNNs for other conditions, such as Diabetic Retinopathy, glaucoma, and normal images, indicates that these conditions might present more nuanced challenges. Specifically, the superior performance of *CNN A* in predicting glaucoma raises questions about the specific architectural elements contributing to this success and prompts further investigation into tailoring architectures for optimal performance across different disease classes.

4) F1 Score: CNN B's exceptional performance in detecting cataracts may be linked to its capacity for capturing intricate features [29], as evidenced by its higher sensitivity and specificity. The balanced F1 scores for diabetic retinopathy across all CNNs suggest their robustness in achieving accurate positive predictions with relatively low false positives (Figure 8). In glaucoma detection, *CNN B*'s superior F1 score hints at its effectiveness in handling the complexities of this class, potentially due to its architectural nuances.

5) Negative Prediction Value: In the case of cataract images, where the NPV is relatively uniform across CNNs, *CNN B* and *CNN C* might exhibit a subtle edge due to architectural features enabling a more nuanced recognition of true negatives (Figure 9). For retinopathy, the consistently high NPV values for all three CNNs suggest their shared proficiency in correctly predicting negative instances within this class, possibly stemming from their common architectural elements. The superior NPV of *CNN B* in glaucoma detection points to its ability to avoid false positives, likely linked to the network's architectural adjustments, such as an additional convolutional layer. In the case of normal images, where NPV values are uniformly high, the slight variations could be indicative of nuanced architectural differences influencing their performance in correctly identifying negative instances.

E. Optimal Epochs

The goal of finding optimal number of epochs is to maximize the learning capabilities of our model and to avoid the over fitting when tested with our validation data. To achieve this we tested the model's performance over multiple sizes of 5, 10, 15, and 20 and plotted the difference in testing and validation accuracy. As shown in Figure 10, it is seen that the least amount of over fitting occurs around 10 epochs, while still yielding a good accuracy result. By fine tuning this aspect of our CNN model, we are able to complete efficient model training and preserve the accuracy of our model to unseen data thus producing optimal generalization.

F. Confusion Matrix

When looking at the confusion matrix (Figure 11) in Figure 11, many important details can be identified. It allows us to identify our model and where it is becoming confused, and how we can further improve the neural network. For example, the most commonly incorrectly classified patterns are cataracts with normal, along with glaucoma and normal. This suggests that glaucoma is a common issue, and it may be worth it to look into pre-processing the data differently. Overall, the confusion matrix backs the model for confidently identifying the correct eye disease classification most of the time, however there is improvement to be made off the commonly incorrectly guessed information.

NOMENCLATURE

adam	Adaptive Moment Estimation
API	Application Programming Interface
CAD	Computer-Aided Diagnosis
CNN	Convolutional Neural Network
F1 Score	Harmonic Mean of Precision and Sensitivity
HRF	High-Resolution Fundus
IOP	Intraocular Pressure
NPV	Negative Predictive Value
OCT	Optical Coherence Tomography
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
SVM	Support Vector Machine
TPU	Tensor Processing Units

REFERENCES

[1] Prevalence of blindness and visual impairment - who. World Health Organization. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment#:~:text=Key%20facts,are%20refractive%20errors%20and%20cataracts

[2] American Academy of Ophthalmology. (2021) Eye examination. Accessed on 2023-09-30. [Online]. Available: https://www.aao.org/eye-health/tips-prevention/eye-exams-101

[3] D. Pavan-Langston, "Chapter 1 - overview of the eye," in *Manual of Ocular Diagnosis and Therapy*, 5th ed., 2013.

[4] National Eye Institute. (2021) Tonometry. [Online]. Available: https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/glaucoma/tonometry

[5] American Academy of Ophthalmology. (2021) Ophthalmoscopy.

[6] A. Saini, K. Guleria, and S. Sharma, "An efficient deep learning model for eye disease classification," in *2023 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, vol. 6, 2023, pp. 1–6.

[7] M. Mishra. (2020) Convolutional neural networks, explained. [Online]. Available: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

[8] D. Helen and S. Gokila, "Eyenet: An eye disease detection system using convolutional neural network," in *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, 2023, pp. 839–842.

[9] B. Krishnamurthy. (2022) An introduction to the relu activation function. [Online]. Available: https://builtin.com/machine-learning/relu-activation-function

[10] P. K. Neha Gour. (2022) Ocular diseases classification using a lightweight cnn and class weight balancing on oct images. [Online]. Available: https://link.springer.com/article/10.1007/s11042-022-13617-1

[11] S. Muchuchuti and S. Viriri, "Retinal disease detection using deep learning techniques: A comprehensive review," *Journal of Imaging*, vol. 9, no. 4, p. 84, Apr 2023. [Online]. Available: http://dx.doi.org/10.3390/jimaging9040084

[12] D. Selvathi and K. Suganya, "Support vector machine based method for automatic detection of diabetic eye disease using thermal images," in *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, 2019, pp. 1–6.

[13] X. Meng, X. Xi, L. Yang, G. Zhang, Y. Yin, and X. Chen, "Fast and effective optic disk localization based on convolutional neural network," *Neurocomputing*, vol. 312, pp. 285–295, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218307112

[14] B. K. Triwijoyo, B. S. Sabarguna, W. Budiharto, and E. Abdurachman, "2 - deep learning approach for classification of eye diseases based on color fundus images," in *Diabetes and Fundus OCT*, ser. Computer-Assisted Diagnosis, A. S. El-Baz and J. S. Suri, Eds. Elsevier, 2020, pp. 25–57. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128174401000024

[15] W. Fan, R. Shen, Q. Zhang, J.-J. Yang, and J. Li, "Principal component analysis based cataract grading and classification," in *2015 17th International Conference on E-health Networking, Application Services (HealthCom)*, 2015, pp. 459–462.

[16] M. M. M. A. R. R. J. . J. O. Marouf, A. A. (2022) An efficient approach to predict eye diseases from symptoms using machine learning and ranker-based feature selection methods. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9854513/

[17] S. S. A. Naser and O. A. A. Zaiter, "An expert system for diagnosing eye diseases using clips," *Journal of Theoretical and Applied Information Technology*, vol. 4, no. 10, pp. 923–930, 2008.

[18] S. T. Y. S. N. T. T. K. T. N. H. Y. Guangzhou An, Kazuko Omodaka and M. Akiba, "Comparison of machine-learning classification models for glaucoma management," 2018.

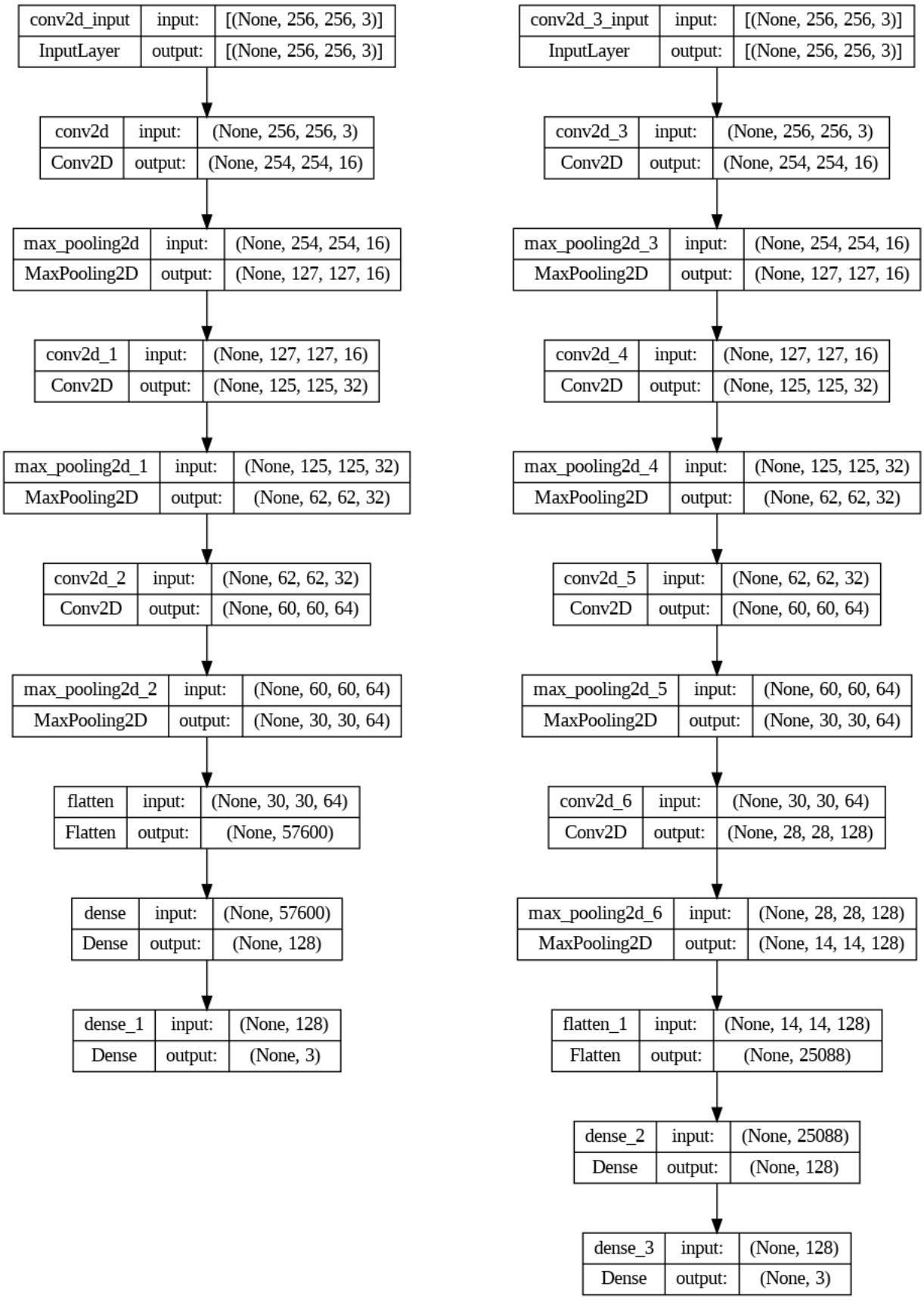
[19] "Convolutional Neural Network (CNN) — TensorFlow Core — tensorflow.org," https://www.tensorflow.org/tutorials/images/cnn, [Accessed 06-12-2023].

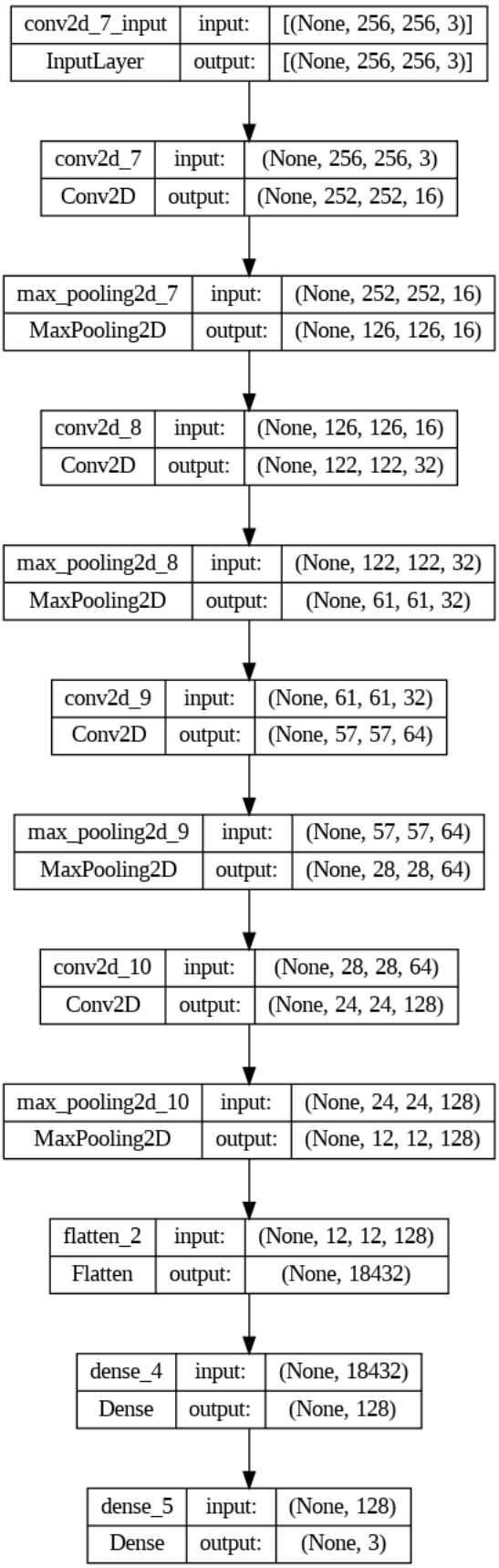
[20] "Softmax function - Wikipedia — en.wikipedia.org," https://en.wikipedia.org/wiki/Softmax_function, [Accessed 06-12-2023].

[21] A. Mishra, "Metrics to evaluate your machine learning algorithm," *Towards Data Science*, February 24 2018, published in Towards Data Science, 7 min read. [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

[22] S. Cristina, "Plotting the training and validation loss curves for the transformer model," *Machine Learning Mastery*, January 6 2023. [Online]. Available: https://machinelearningmastery.com/plotting-the-training-and-validation-loss-curves-for-the-transformer-model/
[23] "ML Metrics: Sensitivity vs. Specificity - DZone — dzone.com," https://dzone.com/articles/ml-metrics-sensitivity-vs-specificity-difference, [Accessed 07-12-2023].
[24] S. Saha, ashukid, and Leevo, "In cnn, why do we increase the number of filters in deeper convolution layers for complex images?" https://datascience.stackexchange.com/questions/55545/in-cnn-why-do-we-increase-the-number-of-filters-in-deeper-convolution-layers-fo, 2023, data Science Stack Exchange.
[25] J. Brownlee, "A Gentle Introduction to Early Stopping to Avoid Overtraining Neural Networks - MachineLearningMastery.com — machinelearningmastery.com," https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/, [Accessed 06-12-2023].
[26] "Epochs, Batch Size, Iterations - How they are Important — sabrepc.com," https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations, [Accessed 06-12-2023].
[27] "Convergence — machine-learning.paperspace.com," https://machine-learning.paperspace.com/wiki/convergence, [Accessed 06-12-2023].
[28] F. H. Reith and B. A. Wandell, "A convolutional neural network reaches optimal sensitivity for detecting some, but not all, patterns," *Psychology Department, Stanford University*, 2023, contact information: Fabian.H.Reith@gmail.com, wandell@stanford.edu.
[29] "Understanding and Applying F1 Score: A Deep Dive with Hands-On Coding — arize.com," https://arize.com/blog-course/f1-score/, [Accessed 06-12-2023].

APPENDIX


conv2d_input | input: | [(None, 256, 256, 3)]
InputLayer | output: | [(None, 256, 256, 3)]
conv2d | input: | (None, 256, 256, 3)
Conv2D | output: | (None, 254, 254, 16)
max_pooling2d | input: | (None, 254, 254, 16)
MaxPooling2D | output: | (None, 127, 127, 16)
conv2d_1 | input: | (None, 127, 127, 16)
Conv2D | output: | (None, 125, 125, 32)
max_pooling2d_1 | input: | (None, 125, 125, 32)
MaxPooling2D | output: | (None, 62, 62, 32)
conv2d_2 | input: | (None, 62, 62, 32)
Conv2D | output: | (None, 60, 60, 64)
max_pooling2d_2 | input: | (None, 60, 60, 64)
MaxPooling2D | output: | (None, 30, 30, 64)
flatten | input: | (None, 30, 30, 64)
Flatten | output: | (None, 57600)
dense | input: | (None, 57600)
Dense | output: | (None, 128)
dense_1 | input: | (None, 128)
Dense | output: | (None, 3)
conv2d_3_input | input: | [(None, 256, 256, 3)]
InputLayer | output: | [(None, 256, 256, 3)]
conv2d_3 | input: | (None, 256, 256, 3)
Conv2D | output: | (None, 254, 254, 16)
max_pooling2d_3 | input: | (None, 254, 254, 16)
MaxPooling2D | output: | (None, 127, 127, 16)
conv2d_4 | input: | (None, 127, 127, 16)
Conv2D | output: | (None, 125, 125, 32)
max_pooling2d_4 | input: | (None, 125, 125, 32)
MaxPooling2D | output: | (None, 62, 62, 32)
conv2d_5 | input: | (None, 62, 62, 32)
Conv2D | output: | (None, 60, 60, 64)
max_pooling2d_5 | input: | (None, 60, 60, 64)
MaxPooling2D | output: | (None, 30, 30, 64)
conv2d_6 | input: | (None, 30, 30, 64)
Conv2D | output: | (None, 28, 28, 128)
max_pooling2d_6 | input: | (None, 28, 28, 128)
MaxPooling2D | output: | (None, 14, 14, 128)
flatten_1 | input: | (None, 14, 14, 128)
Flatten | output: | (None, 25088)
dense_2 | input: | (None, 25088)
Dense | output: | (None, 128)
dense_3 | input: | (None, 128)
Dense | output: | (None, 3)


conv2d_7_input | input: | [(None, 256, 256, 3)]
InputLayer | output: | [(None, 256, 256, 3)]
conv2d_7 | input: | (None, 256, 256, 3)
Conv2D | output: | (None, 252, 252, 16)
max_pooling2d_7 | input: | (None, 252, 252, 16)
MaxPooling2D | output: | (None, 126, 126, 16)
conv2d_8 | input: | (None, 126, 126, 16)
Conv2D | output: | (None, 122, 122, 32)
max_pooling2d_8 | input: | (None, 122, 122, 32)
MaxPooling2D | output: | (None, 61, 61, 32)
conv2d_9 | input: | (None, 61, 61, 32)
Conv2D | output: | (None, 57, 57, 64)
max_pooling2d_9 | input: | (None, 57, 57, 64)
MaxPooling2D | output: | (None, 28, 28, 64)
conv2d_10 | input: | (None, 28, 28, 64)
Conv2D | output: | (None, 24, 24, 128)
max_pooling2d_10 | input: | (None, 24, 24, 128)
MaxPooling2D | output: | (None, 12, 12, 128)
flatten_2 | input: | (None, 12, 12, 128)
Flatten | output: | (None, 18432)
dense_4 | input: | (None, 18432)
Dense | output: | (None, 128)
dense_5 | input: | (None, 128)
Dense | output: | (None, 3)