Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

CataractAsist: Convolutional Neural Network-Based Early Detection System for Cataracts

Nazhif Teggar Ranov[1)]*, **Rinto Priambodo**[2)]
[1,2)]Universitas Pembangunan Jaya, Computer Science Program, Indonesia
[1)]nazhif.teggarranov@student.upj.ac.id, [2)]rinto.priambodo@upj.ac.id

Submitted :Nov 30, 2025 | **Accepted** : Dec 8, 2025 | **Published** : Jan 04, 2026

Abstract: Cataract disease is one of the leading causes of blindness worldwide, especially in developing countries with limited access to healthcare facilities. However, most existing automated detection systems are limited to binary classification, failing to identify specific severity stages. To address this gap, this study develops a novel automated cataract detection system that not only performs three-class grading (Normal, Mature, and Immature) but is also deployed as a web-based application for broader accessibility. Utilizing the "Senile Cataract" dataset from Kaggle, the methodology involves image pre-processing, data augmentation, and feature extraction using the VGG16 model via transfer learning. Experimental results demonstrate that the proposed model achieved 95% accuracy, significantly outperforming the ResNet architecture in comparative benchmarks. While confusion matrix analysis indicates slight misclassification between Immature and Mature classes, the model attained a high F1-score of 0.96 for the Normal class. In conclusion, this study contributes a highly accurate, granular classification system that outperforms standard architectures, offering significant potential as an accessible, automatic early diagnosis tool for eye diseases.

Keywords: Senile Cataract; Convolutional Neural Network; Early Detection; Medical Image Processing; Image Classification

INTRODUCTION

Global Burden & Clinical Urgency Visual impairment constitutes a critical global health burden, with the World Health Organization estimating that over 2.2 billion people suffer from vision impairment worldwide, where at least 1 billion cases could have been prevented (Steinmetz et al., 2021). Among these, cataracts remain the leading cause of blindness, particularly in low- and middle-income countries (LMICs), accounting for approximately 15.2 million cases of blindness globally (Flaxman et al., 2017). In developing nations, the scarcity of ophthalmologists relative to the population creates a significant diagnostic bottleneck. For instance, studies indicate that low-income regions may have as few as 2.7 ophthalmologists per million population, compared to over 79 in high-income regions (Resnikoff et al., 2012). This uneven distribution leaves remote areas underserved, often delaying diagnosis until the disease progresses to irreversible stages. Consequently, there is an urgent need for automated, accessible diagnostic tools to bridge the gap between high-risk populations and early clinical intervention.

Technological Evolution & State-of-the-Art The advent of Artificial Intelligence (AI), specifically Deep Learning (DL), has revolutionized medical imaging analysis. Convolutional Neural Networks (CNNs) have established themselves as the state-of-the-art method for ophthalmological diagnosis, outperforming traditional image processing techniques in feature extraction and pattern recognition (Litjens et al., 2017). Recent studies have employed advanced architectures such as ResNet, InceptionV3, and EfficientNet to detect ocular diseases with high accuracy (Tan & Le, 2019; Ting et al., 2019). These models have demonstrated exceptional capability in identifying pathological features from fundus images and slit-lamp photography, offering a potential solution to the shortage of human experts.

Research Gap Despite these technological advancements, serious limitations persist in current literature and implementation. First, the majority of existing studies focus on binary classification (Cataract vs. Normal). This oversimplification is clinically insufficient because surgical decisions depend heavily on the severity grading, specifically distinguishing between Immature and Mature cataracts (Khan et al., 2021). Second, while newer architectures like EfficientNet offer parameter efficiency, many high-performing models remain confined to computational experiments ("black-box" models) without deployment in practical, accessible frameworks. Third, medical datasets typically suffer from inherent challenges such as class imbalance, noise, and varying illumination

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

conditions, which degrade model performance in real-world scenarios—a challenge often overlooked in standardized benchmark studies (Esteva et al., 2017). Therefore, a distinct gap exists for a system that not only utilizes robust feature extraction for multi-stage classification but also integrates seamlessly into a telemedicine-ready web platform.

Proposed Approach & Justification To address these gaps, this study proposes a CNN-based system utilizing the VGG16 architecture for the three-class classification of cataracts (Normal, Immature, Mature). Although deeper architectures exist, VGG16 is selected for its proven efficacy in capturing detailed textural features essential for differentiating lens opacity stages, while maintaining architectural simplicity for stable deployment (Simonyan & Zisserman, 2015). Furthermore, this research employs data augmentation and transfer learning strategies to mitigate the issues of limited datasets and overfitting.

Objective & Contribution The primary objective of this research is to develop and validate a robust CNN model capable of accurate cataract grading and to deploy this model via a web-based interface to democratize access to eye care. This study offers two key contributions: (1) providing a clinically relevant three-class grading model that assists in prioritizing surgical urgency, and (2) bridging the gap between theoretical AI performance and practical application through a deployed telemedicine tool, specifically designed to support early detection in resource-constrained regions.

LITERATURE REVIEW

Cataract

Cataracts are one of the most common eye disorders, characterized by the formation of opacity in the lens, which causes a decrease in vision. If not treated immediately, this condition can lead to permanent blindness. In healthy eyes, the lens is clear so that light can be focused optimally on the retina. However, degenerative changes or other contributing factors can cause the lens to become cloudy, which seriously impacts visual function. Globally, cataracts are the leading cause of blindness, including in Indonesia, where this disease contributes to approximately 48% of total blindness cases (Firdaus et al., 2022; Ganokratanaa et al., 2023).

The risk factors for cataracts are diverse, ranging from advanced age, genetic history, exposure to ultraviolet rays, systemic diseases, to unhealthy lifestyles such as smoking or the use of certain drugs such as corticosteroids. Infections during pregnancy can also trigger cataracts in babies. Over time, the eye lens undergoes structural changes that interfere with its ability to refract light, causing the image received by the retina to become blurred (Firdaus et al., 2022).

Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is one of the architectures in deep learning that is specifically designed to handle visual data such as digital images. CNNs consist of several main layers, including the convolution layer (Conv2D), which extracts spatial characteristics from images; the pooling layer (MaxPooling2D), which is used to reduce data dimensions in order to reduce calculation complexity; and the Flatten layer, which converts two-dimensional data into one-dimensional vectors so that it can be processed by the next layer. In addition, there is a Dense layer that performs classification. Non-linear activation functions such as ReLU are commonly used in hidden layers, while Sigmoid functions are often applied in output layers for two-class classification (Rismayani et al., 2022). CNNs have proven to be very effective in visual pattern recognition tasks, including cataract detection. One study by Firdaus et al. (2022) developed a web-based CNN model to distinguish between images of normal eyes and eyes with cataracts. The images used were processed to a size of 200×200 pixels before being trained on the CNN model. The model showed high accuracy, reaching 99.74% on the training data and 93.33% when tested with new data. In addition, the web-based implementation allows users to upload images independently and obtain classification results in real-time, making this system easily accessible and practical for use in early cataract detection.

Resnet Architecture

ResNet (Residual Neural Network) is one of the most influential deep convolutional network architectures in the development of CNN-based object detection systems. This architecture introduces the concept of residual blocks with shortcut connections that allow information to flow through multiple layers without degradation, thereby overcoming the vanishing gradient problem that commonly occurs in deep networks. In the context of object detection, ResNet is widely used as a feature extractor or backbone, including in popular algorithms such as Faster R-CNN, RetinaNet, and DETR. One of its implementations, ResNet-101 as the backbone in RetinaNet, was able to achieve a mean Average Precision (mAP) of 53.1% on the MS COCO dataset, demonstrating competitive performance compared to other models such as YOLOv3 and SSD.

The main advantage of ResNet lies in its ability to extract complex spatial features and its compatibility with architectures such as Feature Pyramid Network (FPN) to improve multi-scale detection. However, high computational consumption is one of the main limitations of this architecture. To be efficiently applied to real-

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

time systems or devices with limited resources, additional optimization strategies are required, such as model compression, pruning, or the use of lighter architecture variants (Arkin et al., 2022).

VGG16 Architecture

VGG-16 is one of the Convolutional Neural Network (CNN) architecture models presented by Simonyan & Zisserman (2014). This structure has 16 layers, consisting of 13 convolutional layers, fully connected layers, and 1 output layer that uses the Softmax activation function. A distinctive feature of VGG-16 is the consistent use of 3x3 convolution kernels in each convolutional layer, as well as the repeated use of 2x2 pooling layers, which form a simple yet highly efficient network structure for extracting spatial information from images. Although not the latest model, VGG-16 is still often used in various image classification and transfer learning applications due to its stability and ease of implementation. VGG-16 was applied to identify damage in coffee beans and achieved a training accuracy of 96.17% and a testing accuracy of 87.30% for positive classification. These results show that VGG-16 is still an appropriate and efficient option for classifying images with two categories, especially for data with striking visual features.

Flask

Flask is a Python-based microframework designed to provide high flexibility in web application development. Flask only includes core features such as routing, request management, and the Jinja2 template engine, allowing developers to freely add extensions as needed, such as SQLAlchemy for ORM, Flask-Login for authentication, and Flask-WTF for form validation. Due to its lightweight and modular design, Flask is well-suited for prototyping and small to medium-sized applications. In addition, Flask loosely adopts the Model-View-Controller (MVC) design pattern, giving developers complete freedom in structuring their projects. Flask was used to build a social networking application that allows users to follow and interact with each other. The results show that Flask offers ease of configuration, comprehensive documentation, and ease of deployment to platforms such as Heroku, making it an ideal choice for developers who prioritize speed and full control over the application development flow.

Previous Research

The previous achievements section is used as a reference to strengthen the researcher's arguments and as a reference in conducting research and designing systems. Researchers publish their research results in previously published journals. The following are some journals that have differences but are still relevant to the issues discussed and the algorithms applied in the research.

Table 1. Previous Research

Author (Year)	Method	Dataset	Accuracy	Limitations
(Andreas et al., 2023)	CNN (InceptionV3)	400 fundus images (300 normal, 100 cataract), 70/30 split	97% without augmentation; 100% with augmentation	Small dataset; only 2 classes; no comparison with other CNN architectures
(Nurona Cahya et al., 2021)	CNN (AlexNet)	610 fundus images, 4 classes (normal, cataract, glaucoma, retinal disorder)	98.37%	Some cataract images are still misclassified as normal; possible class overlap
(Ganokratanaa et al., 2023)	Custom CNN (3 conv, 3 pooling, 2 fully connected) + 5-fold CV	600 eye images (balanced normal vs cataract)	97% accuracy; 95% precision; 100% recall; 97% F1-score	Still produces false negatives on cataract images that appear similar to normal
(Firdaus et al., 2022)	Black-box testing with diversity metrics (GD, SD, NCD) using VGG-16 features	CIFAR-10, MNIST, Fashion-MNIST, SVHN; 5 CNN models	Accuracy not the main focus	Uses generic vision datasets; not eye-disease images; focuses on correlation analysis, not detailed classification metrics
(Aghababaeyan et al., 2023)	Black-box testing for CNN-based eye-disease detection system	Image-based eye-disease detection system (dataset	No explicit accuracy reported	Focuses on testing strategy; dataset details and performance

*name of corresponding author

		not clearly described)		metrics are not clearly provided
(Shidik et al., 2024)	Transfer learning: ResNet50, DenseNet121, MobileNet with augmentation	1,034 fundus images (normal, cataract, glaucoma, diabetic retinopathy)	ResNet50: 92.27%; DenseNet121: 91.79%; MobileNet: 89.86%	ResNet50 gives highest accuracy but requires high compute; MobileNet is efficient but less accurate
(Richard et al., 2022)	YOLOv3, YOLOv4 and CNN object detection	Various domains (surveillance, UAV, autonomous vehicles, livestock)	Competitive performance; no single consolidated accuracy	Performance depends heavily on hardware and environment; not specific to cataract or fundus images
(Arkin et al., 2022)	Survey: CNN (Faster R-CNN, YOLO, SSD) and Transformer-based (DETR, ViT, Swin)	Public benchmarks: PASCAL VOC, MS-COCO, Open Images	mAP, FPS, IoU, recall reported from existing literature	Pure survey; no new model; Transformer-based methods require large datasets and heavy compute; not focused on medical imaging
(V et al., 2022)	Review of real-time object detection with YOLO and CNN	Literature using PASCAL VOC, MS-COCO	Example results: YOLOv4 mAP ≈ 87.48%; YOLOv3-Tiny mAP ≈ 69.79% @ 28 FPS	Accuracy and speed depend on hardware; YOLO has localization errors and accuracy drops after quantization
(Ghimire, 2020)	Implementation study of Flask web framework	No ML dataset; simple social-network web application	Not applicable (no ML model)	Focuses on one small web app; Flask has limited built-in features (security, DB migration require manual handling)
(Rafli et al., 2024)	White-box testing (basis-path, Cyclomatic Complexity)	Sistem web "A Laundry" (login & employee)	Tidak relevan (software testing)	Hanya 2 modul diuji; tidak menunjukkan performa pada sistem lebih kompleks

Based on existing literature, cataract detection research using CNN still shows significant gaps. Although models such as ResNet and MobileNet have been used, previous studies generally focus on binary classification (Normal vs. Cataract) or ignore the integration of models into systems that are easily accessible to non-technical users. Therefore, this study aims to fill these gaps by integrating the VGG16 transfer learning architecture, which has been fine-tuned for three-class cataract detection (Normal, Immature, and Mature), and implementing it as a portable, publicly accessible web-based system. The differentiation of three-class classification and the provision of a web platform is what uniquely positions this research from previous studies.

METHOD

Hardware & Software Requirements

The hardware used in this study includes a laptop as the main data processing unit. This laptop is used to run all stages of the system, from image pre-processing, feature extraction using the Convolutional Neural Network (CNN) model, to the classification and validation of results. The dataset in the form of eye images was downloaded from the Kaggle platform and processed locally on this device. In its implementation, the laptop acts as a computing center that handles model training and testing, while also running a web-based system designed to automatically detect and classify eye conditions (normal or cataracts). The following are the hardware specifications used for training and testing in this study:

- CPU: Intel(R) Core(TM) i5-11400H 2.70GHz
- GPU: Nvidia GeForce RTX 3050 Laptop GPU

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v10i1.15642 p-ISSN : 2541-044X

- RAM: 16Gb 8x2

On the software side, this research uses Visual Studio Code as the main development environment, combined with several Python-based programming libraries. The libraries used include TensorFlow and Keras to build and train CNN models, scikit-learn (SKLearn) to support evaluation and classification metrics, Matplotlib for model performance visualization, and NumPy for numerical data processing. All of these software components are integrated into a single system that functions to read images, extract important features, perform classification, and present the results in a web interface that is easily accessible to users.

Cataract Detection System Framework

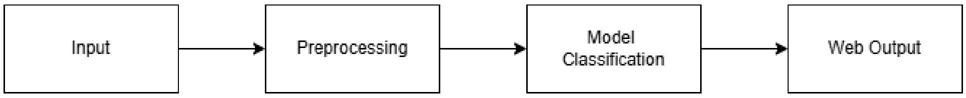

Fig. 1 Diagram Block System Cataract Detection

This early cataract detection system operates through a sequential workflow that begins with the input of eye images and produces a web diagnosis (Web Output)1. The first stage is Input, where digital images of the eye are uploaded by users via a web interface22. These images then undergo Preprocessing, which is essential for standardizing and normalizing the data. Preprocessing includes adjusting the image dimensions (such as to a standard size of 150 × 150) and normalizing the pixel values from a range of 0-255 to 0-1, ensuring that the data is ready and consistent for processing by the model.

Once the data is prepared, the next stage is Model Classification, which is the core of the system. This stage uses the Convolutional Neural Network (CNN) VGG16 architecture through the Transfer Learning4 approach. This model functions as an intelligent feature extractor, recognizing important visual patterns in eye images to classify them into three categories: Normal, Immature Cataract, or Mature Cataract5. The training and testing of this model has shown excellent performance, achieving an overall accuracy of 95%.

Finally, the classification results from the model are converted into actionable information at the Web Output stage. The system is implemented as a Flask-based web application that allows users to receive analysis results instantly and real-time after uploading images. The web output displays the diagnosis (e.g., Your Eye Has Been Diagnosed with Cataract) and is accompanied by AI-based recommendations, positioning this system as a fast, accurate, and easily accessible early diagnosis tool.

Proposed System Design

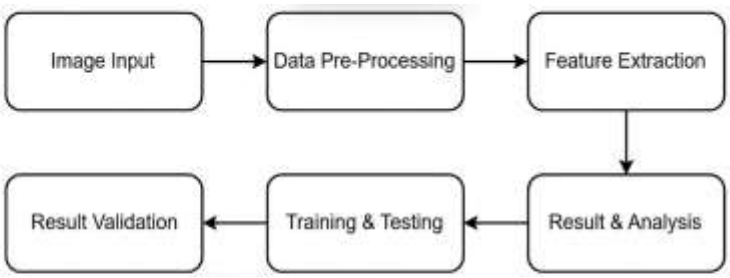

Fig. 2 System Block Diagram

Based on the block diagram in Figure 1, this study consists of six main stages, starting with image input from the Senile Cataract Dataset obtained through the platform (https://www.kaggle.com/datasets/rifdana/dataset-katarak-sinilis). This Kaggle Dataset provides eye images classified as normal and senile cataracts, which are used as training and test data in this study. The next stage is pre-processing, which aims to improve image quality and prepare it for further analysis. This process includes resizing images to a standard size and normalizing pixel values to suit the model input requirements.

After that, the processed images enter the feature extraction stage using a Convolutional Neural Network (CNN) architecture, which is capable of recognizing important visual patterns from eye images. The features obtained are then selected at the feature selection stage to extract the most relevant and significant features for classification. Next, the training and testing stages are carried out using a CNN model with a VGG-16 architecture , in order to produce the best model for distinguishing between normal eye images and those with senile cataracts. The final stage is to validate the results using evaluation metrics such as accuracy, precision, and recall, to measure

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

the model's performance against the test data and assess the effectiveness of the system in automatically detecting cataracts.

Web-Based System Workflow

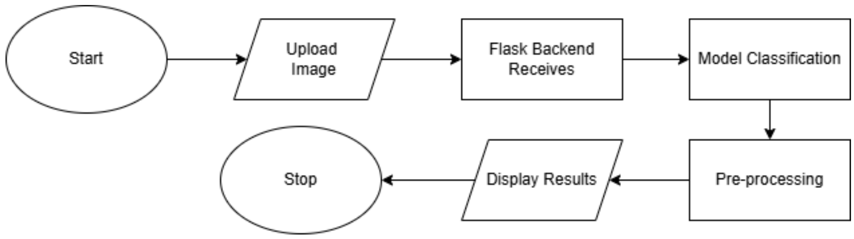

Fig. 3 Web-Based System Workflow

The overall architecture of the cataract detection system, as outlined in the System Block Diagram, defines a linear process from image input to diagnostic output. The flow begins with the Input of the eye image, which immediately proceeds to the Preprocessing stage where the data is standardized for the model's consumption. The prepared data is then fed into the Model Classification block, which performs the automated diagnosis. The process concludes with the Web Output, where the diagnostic result is presented to the user. Supporting this output is the Web Flowchart, which details the user interaction flow: the process Starts, the user Uploads Image, and the Flask Backend Receives the file. The core classification is executed by the CNN model, and the result is delivered to the user for display (Display Results), bringing the process to an End.

The classification model itself utilizes an adapted VGG16 Architecture based on a Transfer Learning approach. The structure is divided into two key segments: the Pretrained Layer (convolutional base) and the Fully Connected Layer (classification head). The convolutional base, which handles initial feature extraction, is largely set to Non-Trainable to leverage the features learned from the ImageNet dataset, thus saving training time. However, the final convolutional blocks are marked as Trainable to allow for fine-tuning the deep features to better suit the specific domain of eye medical images. The subsequent Fully Connected Layer is entirely Trainable and includes custom layers: a Flatten layer, Dense layers for final computation, and the Output layer that yields the final classification into one of the three defined classes: Normal, Immature, or Mature Cataract.

Hyperparameter Configuration

In the training process of the VGG16-based Convolutional Neural Network (CNN), the selection of hyperparameters plays a pivotal role in achieving optimal convergence and mitigating the risk of overfitting. The configuration used in this study was strategically designed to strike a balance between computational efficiency and classification accuracy.

The base architecture utilizes the VGG16 model, with input images standardized to dimensions of 150 x 150 pixels across three color channels (RGB). For weight optimization, the Adam Optimizer is employed with a learning rate of 0.001, selected for its adaptive learning capabilities and computational efficiency. The training process is set to run for a maximum of 100 epochs with a batch size of 32, allowing for frequent weight updates while maintaining memory stability.

To ensure the model generalizes well to unseen validation data, a Dropout regularization technique with a rate of 0.5 is applied to the Fully Connected layers. This mechanism randomly deactivates 50% of the neurons during the training phase to prevent co-adaptation. Furthermore, given the multi-class nature of the specific problem (Normal, Immature Cataract, and Mature Cataract), Categorical Cross-entropy is utilized as the loss function. The complete details of the hyperparameter configuration are presented in Table 2.

Table 2. Hyperparameter Configuration

Parameter	**Configuration**
Arsitektur Dasar	VGG16 (Pretrained ImageNet)
Input Shape	150X150X3
Optimizer	Adam
Learning Rate	0.001

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Batch Size	32
Epoch	100 (with Early Stopping)
Dropout Rate	0.5
Loss Function	Categorical Cross-entropy

Data Collection

In this study, the author used a dataset downloaded from the Kaggle platform titled "Senile Cataract Dataset," which can be accessed via the following link: https://www.kaggle.com/datasets/rifdana/dataset-katarak-sinilis. This dataset consists of a number of eye images classified as normal and senile cataract , which are used for training and testing an automatic detection model based on Convolutional Neural Network (CNN). This dataset was selected based on its relevance to the research objective, namely the development of an early detection system for eye diseases based on digital images.

The images in this dataset have a high enough resolution to support accurate visual detection. All image files are available in .jpg format and are systematically organized into several folders representing each class or category. Although this dataset does not include metadata as detailed as certain medical datasets such as Sinus Cataract, the image storage structure based on labels and good visual quality is sufficient to support image processing-based classification.

The use of this dataset is highly relevant for the development of image-based disease detection systems because it provides real visual examples of cataract medical conditions, which are important for training models to recognize the typical patterns of senile cataracts.

Pre-processing

The pre-processing stage is a very important first step in a CNN-based cataract detection system. At this stage, image data is prepared and normalized to match the format and scale that can be processed by the model. One of the pre-processing methods used is the normalization of image pixel values by adjusting the pixel scale from 0–255 to 0–1 using the rescale=1./255 parameter. This normalization is performed for both training data (train_datagen) and test data (test_datagen) to improve model training efficiency and maintain neural network learning stability.

The training data is automatically divided into two subsets, namely the training subset and the validation subset, with a ratio of 80:20. This division is done by utilizing the validation_split=0.2 parameter in the ImageDataGenerator object. Then, the images used in training and validation are taken from the same directory (train_dir) and read using the flow_from_directory method. The target_size=(150, 150) parameter is used to standardize the image dimensions so that they can be processed by the CNN model, while batch_size=32 sets the number of images processed in one iteration. The classification mode used is categorical, because the system will classify images into several class categories.

Meanwhile, for the test data, a test_generator object is used, which is formed from the test_dir directory. This test data does not go through the shuffle process (shuffle=False) so that the image sequence remains in accordance with the original label when evaluation is performed. All test images are also normalized and resized in the same way as the training data.

Overall, this pre-processing stage ensures that all images entering the CNN model have a uniform pixel scale, consistent size, and are well divided between training, validation, and testing data. This is very important so that the model can learn from the data optimally and produce accurate predictions in detecting eye conditions, especially cataracts.

Feature Extraction

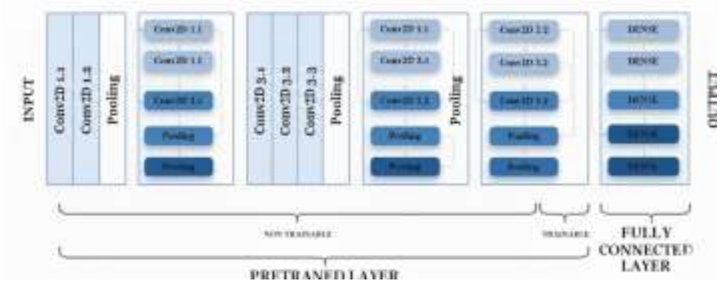

Fig. 4 VGG16 Structure

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

The VGG16 architecture, as utilized in this research, is structured into two main components: the Pretrained Layer (convolutional base) and the Fully Connected Layer (classification head), employing a Transfer Learning strategy. The initial part consists of the deep convolutional layers, which are derived from the original VGG16 model pretrained on the ImageNet dataset. This convolutional base is responsible for extracting rich visual features from the input images, using consistent 3 x 3 convolutional kernels followed by 2 x 2 pooling layers.

In this specific implementation, the convolutional layers are primarily set to Non-Trainable (vgg_base.trainable = False) to preserve the robust, general feature-extraction capabilities learned from ImageNet, significantly saving training time. However, the diagram suggests that the final block of convolutional layers (Conv2D 3.1 to Pooling) is marked as Trainable, indicating a fine-tuning strategy where some of the deep feature layers are updated during training to better suit the specific domain of eye medical images. The entire convolutional base is loaded without the original classification top layer (using the include_top=False option).

Following the convolutional base is the Fully Connected Layer (the new classification head), which is the Trainable part added specifically for the three-class cataract classification task. This section begins with a Flatten layer to convert the 2D feature maps into a 1D vector. This is followed by a Dense layer with 512 units and a ReLU activation function to learn more complex feature combinations. To prevent the model from overfitting, a Dropout layer with a 50% rate is inserted, which randomly deactivates neurons during training. The final Output layer then performs the actual classification into the three defined categories: Normal, Immature, and Mature Cataract, To convert the output logits into probability distributions for each class, the Softmax activation function is applied, as shown in Equation:

$$\partial(z)_i = \frac{e^{zi}}{\sum_{j=1}^{k} e^{zj}}$$

Where z is the input vector, is the number of classes, and sigma(z) is the output probability. During the training process, the discrepancy between the predicted probability and the actual label is measured using the Categorical Cross-Entropy loss function, defined in Equation:

$$L_{CCE} = -\sum_{i=1}^{K} yi \cdot \log\,(\hat{y}i)$$

Where yi is the ground truth label and $\hat{y}i$ is the predicted probability

Image Processing Process

The image processing process aims to enrich the variety of training data without the need to manually add new images, so that the model can learn better and become more reliable in various image conditions. Several transformations applied in this process include normalizing image pixel values through the rescale=1./255 parameter, which changes the pixel range from 0–255 to 0–1. This is important for speeding up the training process and ensuring stability in neural network learning. Furthermore, augmentation is performed by randomly rotating the image up to 40 degrees through rotation_range, as well as shifting the image position horizontally and vertically using width_shift_range and height_shift_range.

In addition, shear_range is used to apply a skew distortion effect to the image, while zoom_range allows the model to see images on a larger or smaller scale. The augmentation process also includes horizontal_flip, which flips the image from left to right to introduce diversity in the appearance of the eye image. The fill_mode='nearest' parameter is used to fill empty areas resulting from the transformation with the nearest pixel values to keep the image intact.

Training Process

To systematically illustrate the operational logic of the proposed method, the detailed training procedure is formalized in Algorithm 1. This algorithm outlines the sequential execution of the model, starting from the initialization of the VGG16 architecture to the iterative optimization process. The workflow requires the preprocessed training set (D_train) and validation set (D_val) as primary inputs, along with defined hyperparameters including a maximum of 20 epochs and a batch size of 32. As depicted in the algorithm, the core mechanism involves a loop that iteratively updates the model weights via backpropagation based on the Categorical Cross-Entropy loss. Crucially, the logic integrates a conditional check for the Early Stopping strategy after each epoch. The algorithm monitors the validation loss (Val_Loss); if no improvement is observed for a consecutive number of epochs defined by the patience parameter (P=3), the training loop terminates prematurely. This logical control flow ensures that the system prioritizes the model with the highest generalization capability—represented by the restored best weights—rather than simply completing all iterations, thereby effectively mitigating the risk of overfitting.

ALGORITHM 1: CNN Training Workflow with Early Stopping

INPUT: Training_Set (D_train), Validation_Set (D_val) Max_Epochs = 20 Batch_Size = 32 Patience (P) = 3

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

```
OUTPUT: Trained_Model (M_best)

BEGIN // 1. Model Initialization Base_Model = Load VGG16(weights='imagenet', include_top=False) Freeze
Base_Model layers

// 2. Add Classification Head
Head = Flatten() + Dense(512, activation='ReLU') + Dropout(0.5) + Dense(3, activation='Softmax')
M = Combine(Base_Model, Head)

// 3. Compile Model
Compile M using:
  Optimizer = Adam(learning_rate=0.001)
  Loss_Function = Categorical Cross-Entropy

// 4. Training Loop
Best_Val_Loss = Infinity
Patience_Counter = 0

FOR epoch = 1 TO Max_Epochs DO:
  // Training Step
  FOR EACH batch (X_batch, y_batch) IN D_train DO:
    Predictions = M.predict(X_batch)
    Loss = Calculate_CrossEntropy(y_batch, Predictions)
    Update M weights via Backpropagation
  END FOR

  // Validation Step
  Val_Loss, Val_Accuracy = M.evaluate(D_val)

  // Early Stopping Logic
  IF Val_Loss < Best_Val_Loss THEN
    Best_Val_Loss = Val_Loss
    Save current weights as Best_Weights
    Patience_Counter = 0
  ELSE
    Patience_Counter = Patience_Counter + 1
    IF Patience_Counter >= P THEN
      PRINT "Early Stopping Triggered"
      Restore Best_Weights to M
      BREAK Loop
    END IF
  END IF

  PRINT epoch, Loss, Accuracy, Val_Loss, Val_Accuracy
END FOR

RETURN M (with Best_Weights)
END
```

At this stage, the Convolutional Neural Network (CNN) model is trained to recognize and classify eye images based on cataract conditions. The training process is carried out by considering the model's performance on training and validation data to avoid overfitting issues.

The Early Stopping technique from the tensorflow.keras.callbacks library plays a role in automatically stopping the training process if there is no improvement in model performance on the validation data in the last few epochs. The parameter monitor='val_loss' indicates that the training process will be evaluated based on the loss value on the validation data. If there is no improvement in 3 consecutive epochs (patience=3), then training will be stopped. The restore_best_weights=True option will ensure that the model retains the best weights obtained during training.

*name of corresponding author

DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Next, the model is trained using the previously processed data, namely train_generator for training data and validation_generator for validation data. The training process runs for a maximum of 20 epochs, but may stop earlier if the Early Stopping condition is met. After the training process is complete, the model's performance is visualized by plotting the accuracy values on the training and validation data.

Fine Tuning

In the development of Deep Learning research in the 2020–2025 period, Transfer Learning strategies remain the dominant approach to overcoming data scarcity in medical image analysis. Alzubaidi et al. (2021) in their comprehensive review confirmed that the use of pre-trained models significantly reduces the computational load and accelerates model convergence compared to training from scratch, especially on datasets with high variability but limited sample sizes.

Regarding weight adaptation strategies, recent research supports the partial fine-tuning method. Matsoukas et al. (2022) in their study on transferability in medical imaging found that fine-tuning all layers of the network often does not provide a performance improvement commensurate with the risk of catastrophic forgetting, where the model forgets the general features it has learned previously. Therefore, a hierarchical approach of freezing the initial feature extraction blocks (such as Blocks 1–4 in ResNet) and training only the final convolutional block (Block 5) has become a practical standard. This approach allows the model to retain robust basic visual filters, while adapting high-level semantic features to suit the specific characteristics of cataracts Talo et al. (2019).

In addition to the training strategy, modifications to the classifier head architecture were also made for efficiency. The use of Global Average Pooling (GAP) as a substitute for the conventional Flatten layer has been increasingly validated for reducing feature dimensions without losing global spatial information Chowdhury et al. (2020). Furthermore, the addition of a Dense layer with moderate capacity (such as 512 neurons) combined with the Dropout regularization technique remains relevant. Research by Khan et al. (2021) on eye disease classification shows that this configuration is effective in preventing overfitting and improving the model's generalization ability on previously unseen validation data.

Testing Process

The process begins by using the evaluate() function, which provides two key metrics, namely loss and accuracy, on the test data. The loss value indicates the extent of the error made by the model in predicting, while the accuracy value indicates the level of accuracy of the model's predictions. This evaluation provides an overview of the model's performance in processing the test data.

Next, to obtain more in-depth results, predictions are made on the test data using the predict() function. These predictions are then compared with the actual classes (correct labels) to evaluate the model's accuracy in more detail. The predictions generated are probabilities for each class, which are then converted to the most likely class using the np.argmax() function. These predictions are compared with the actual classes to obtain further evaluation metrics, including precision, recall, and f1-score, which are listed in the classification report. This report provides a more detailed analysis of the model's performance on each class, helping to identify classes that are more difficult to recognize.

Result Validation

In the Result Validation stage of the cataract detection system using image processing based on the CNN method, the first step is to make predictions on the test data using the previously trained model. This process begins by utilizing the predict() function in the model to generate predictions from the test data. Each prediction provided by the model is in the form of a probability for each class, which is then processed using np.argmax() to convert the probability into the most likely class label, namely the class with the highest probability value.

After that, the prediction results (y_pred) are compared with the original classes (y_true) found in the test data. The model's performance is evaluated using two main tools: Classification Report and Confusion Matrix. Classification Report provides a complete overview of the model's performance in terms of accuracy, precision, recall, and F1-score for each class. These metrics provide insight into how well the model can classify images correctly, as well as how well the model handles different classes, whether for cataract or normal classification.

Furthermore, to understand the distribution of classification errors, a Confusion Matrix is used. This matrix provides information about correct and incorrect predictions between existing classes. In the visualization of the confusion matrix, we can see the number of correct and incorrect predictions for each class, which is depicted in the form of a heatmap to facilitate interpretation. This heatmap is equipped with numerical annotations on each cell to show the number of related events. These results help us identify areas that need improvement in the model and provide additional insight into which classes are often confused or misclassified.

Additionally, this confusion matrix image is saved using plt.saveImage() for further documentation. Through this evaluation, the model can be assessed in greater depth, and optimization steps can be designed based on the analysis of the classification report and confusion matrix.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Testing Scenario

In this section, the image classification model that has been built using test data is tested. The main objective of this test is to objectively evaluate the model's performance on data that has never been used in the training process ( ). The test is carried out in several stages, namely model evaluation, prediction process, classification report generation, and confusion matrix analysis. Each stage is designed to provide an in-depth understanding of the model's performance in accurately classifying images.

Model Evaluation

The first stage in the testing scenario is to evaluate the model against the test dataset. This evaluation is performed using the built-in model.evaluate() function from the Keras library, which accepts test_generator as a parameter. This function calculates two main metrics, namely loss and accuracy values. The loss value represents the level of error in the model's predictions against the actual data, while accuracy shows the percentage of correct predictions from all test data. Mathematically, the accuracy metric is calculated based on the confusion matrix components using Equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} X100\%$$

Where TP represents True Positives, TN for True Negatives, FP for False Positives, and FN for False Negatives.

Prediction and Evaluation

After the evaluation process is complete, the next step is to make predictions on the test data. The previously trained model is used to predict classification results using the model.predict(test_generator) command. The output of this prediction function is a two-dimensional array representing the probability of each class for each test image. To obtain the final class label for each prediction, the np.argmax() function is used with the axis=1 parameter, which takes the index with the highest probability. This index becomes the prediction label (y_pred). Meanwhile, the actual labels of the test data are obtained from the classes attribute of the test_generator object and stored in the y_true variable. These two variables, y_true and y_pred, will be used to evaluate the prediction accuracy in more detail in the next stage.

Classification Report

A classification report is generated to provide a more comprehensive overview of the model's performance for each class. This process is carried out using the classification_report() function from the sklearn.metrics library. This function requires input in the form of actual labels (y_true), predicted labels (y_pred), and a list of class names obtained from test_generator.class_indices.keys(). This classification report includes several evaluation metrics such as precision, recall, f1-score, and support. Precision measures the accuracy of the model in classifying a class, recall measures how well the model recognizes all data from a class, while f1-score is the harmonic mean of precision and recall. Support shows the number of actual samples from each class. With this report, the model's performance is not only seen from its overall accuracy, but also how well the model is able to recognize and distinguish each existing class.

Confusion Matrix

The final stage of the testing scenario is the creation and analysis of a confusion matrix. This matrix is generated using the confusion_matrix() function from the sklearn.metrics library, which accepts y_true and y_pred as input. The confusion matrix serves to visualize the errors and successes of the model's predictions in a two-dimensional table. Each row of the matrix represents the actual label, while each column represents the predicted label. By looking at the values on the diagonal of the matrix, you can see the number of correct predictions for each class. Meanwhile, the values outside the diagonal indicate incorrect predictions. The confusion matrix is very useful for identifying which classes are often mispredicted, which can be used as a basis for model improvement, such as adjusting the architecture, adding training data, or applying data augmentation techniques.

RESULT

Training Model

After evaluating and testing the classification model using a senile cataract image dataset consisting of 495 samples, the results showed that the model performed quite well. The dataset was evenly divided into three classes, namely Immature, Mature, and Normal, each consisting of 165 test data. Based on the classification report results, the model showed an overall accuracy of 95%. This value reflects that 95% of the total test data was successfully classified into the correct class.

Table 3
Classification report

*name of corresponding author

	Precision	Recall	f-1 Score	Support
Immature	0.99	0.87	0.93	165
Mature	0.92	0.98	0.95	165
Normal	0.94	0.99	0.96	165
Accuracy			0.95	495
Macro Average	0.95	0.95	0.95	495
Average Weight	0.95	0.95	0.95	495

In more detail, Table 1. Classification report explains that the model's performance for each class can be analyzed through precision, recall, and f1-score values. For the Immature class, a precision of 0.99 was obtained, which means that almost all Immature predictions were correct. However, the recall is only 0.87, indicating that the model fails to detect all data that should be included in this class, possibly because some Immature samples are misclassified into other classes. The F1-score for this class is 0.93, which reflects a fairly good balance between precision and recall. For the Mature class, the precision is 0.92 and the recall is very high at 0.98, indicating that almost all images that are truly Mature are successfully recognized by the model. The F1-score is also high, at 0.95. Meanwhile, the Normal class has the highest recall value of 0.99 with a precision of 0.94, which means that most of the predictions and detections of Normal images are done very well, resulting in an F1-score of 0.96.

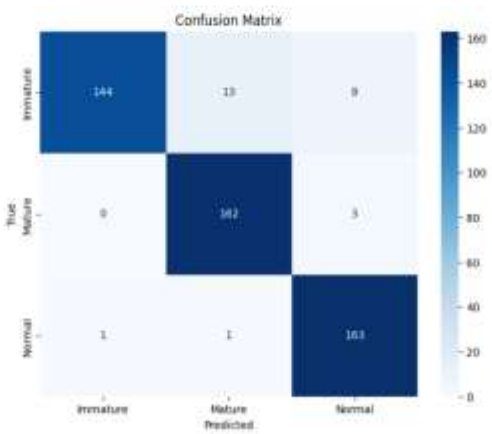

Fig. 5 Confusion Matrix Results

Further analysis using the confusion matrix in Figure 2 shows that out of 165 images in the Immature class, 144 images were classified correctly, while 13 images were classified as Mature and 8 as Normal. For the Mature class, 162 images were predicted correctly, and only 3 images were misclassified as Normal. Meanwhile, from the Normal class, 163 of the 165 images were classified correctly, and one image each was misclassified as Immature and Mature. This matrix indicates that most classification errors occurred between the Immature and Mature classes, which was likely due to the similarity of visual features between the two classes.

Analysis of the model's performance reveals that the most significant classification errors primarily occur between the Immature Cataract and Mature Cataract classes. This conclusion is strongly supported by the confusion matrix, which shows that out of 165 Immature images, 13 were incorrectly predicted as Mature. Consequently, the Immature class recorded the lowest Recall value at 0.87, indicating that the model failed to detect 13% of the samples that were truly Immature, with misclassification into the Mature and Normal classes being the main cause. The high recall for the Mature class (0.98), in contrast, suggests that mature cataracts are visually clearer and easier for the CNN to recognize. The core reason for this dominant error trend is attributed to the similarity of visual features between the Immature and Mature stages, specifically the variation in lens opacity that falls at the classification threshold, making these two categories the most challenging for the model to distinguish. This suggests that improving feature differentiation between Immature and Mature cases is key for future optimization efforts.

Overall, both the Classification Report and Confusion Matrix show that the model built has very satisfactory performance. The macro and weighted average values for precision, recall, and F1 score are all at 0.95, indicating consistent model performance across all classes. These results show that the model is capable of recognizing and

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

distinguishing types of senile cataracts with high accuracy, making it a strong foundation for further development in medical image classification applications, particularly in the automatic detection of cataract stages.

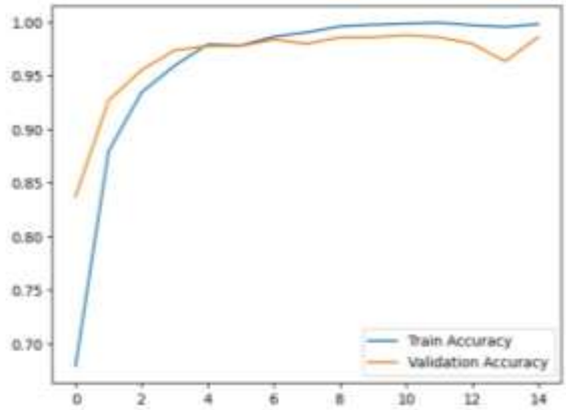

Fig. 6 Train vs Validation Accuracy VGG16

The graph represents the training and validation accuracy of the VGG16 model over 15 epochs. In the early epochs, both accuracies increase rapidly, showing that the model quickly learns essential features from the eye images due to the advantages of transfer learning. After around the fourth epoch, the accuracy stabilizes at a high level, ranging between 0.95 and 1.00 for both training and validation, indicating that the model has effectively captured the necessary patterns for cataract classification. The close alignment of the two curves suggests that the model does not suffer from significant overfitting, as its performance on unseen validation data remains consistently strong. Although there is a slight fluctuation in validation accuracy around epochs 12–13, this variation is minimal and normal in deep learning training. Overall, the results demonstrate that VGG16 performs exceptionally well and is highly reliable for three-class cataract detection.

Table 3. Comparison of VGG16 and ResNet Performance on Cataract Classification

Model	Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
VGG16	Epoch 15	0.9965	0.9858	0.0172	0.0537
ResNet	Epoch 99 (stopped at best epoch 97)	~0.57	~0.64	~0.90	~0.78

VGG16 demonstrates excellent learning behavior. Both training and validation accuracy reach above 98%, and the losses are extremely low. The small gap between training and validation metrics indicates that the model generalizes well and does not suffer from overfitting. These results show that VGG16 is highly effective for this dataset, able to capture visual features of cataract classes with high precision.

On the other hand, ResNet performs noticeably worse. Even after nearly 100 epochs, the training accuracy remains around 0.53–0.57, and validation accuracy stays between 0.63–0.65. The losses also remain high, suggesting that the model struggles to converge. This indicates underfitting and poor feature learning, possibly due to mismatch between the dataset size and the complexity of ResNet, or hyperparameters that do not fit well for this specific task.

Overall, based on the comparison, VGG16 clearly outperforms ResNet in accuracy, stability, convergence speed, and generalization. This makes VGG16 the more suitable model for three-class cataract classification in your system.

Web Application

To facilitate access and expand the scope of use, the Convolutional Neural Network (CNN)-based cataract early detection system has been implemented into a web-based application. This system is designed to provide a fast, efficient, and user-friendly detection process. Users simply upload images of the eye through the main page of the web application, then the system automatically processes the images using a CNN model running on a Flask-based backend. The results of the model's analysis are displayed instantly to the user, including information about the possibility of cataracts and the diagnosis provided by the system.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

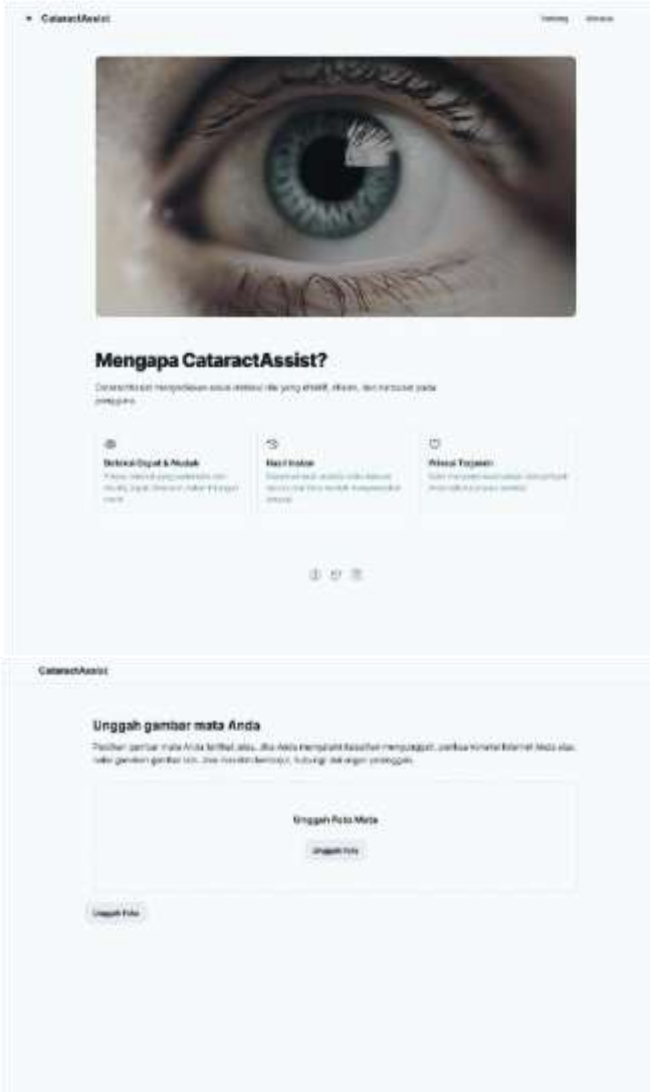

Fig. 7 Web Page Display

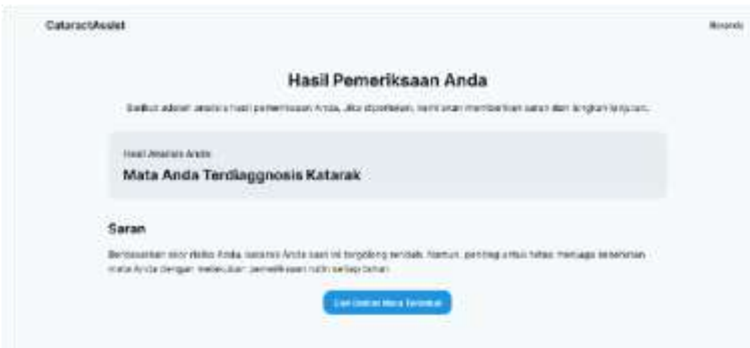

Fig. 8 Web Page Display

As shown in Figure 3 and 4, the application interface is designed to be simple and responsive to provide an intuitive user experience. The application displays detection results in real-time, where users will receive notifications on whether their eyes are diagnosed with cataracts or not, accompanied by artificial intelligence-based recommendations that can help users take the next steps regarding their eye health. The system also ensures data security and privacy by keeping all user information encrypted and not sharing it without consent.
The web-based implementation makes it a more inclusive solution, especially for people in areas with limited access to medical services. In addition, with this approach, the system has the potential to be integrated into the

*name of corresponding author

 e-ISSN : 2541-2019
p-ISSN : 2541-044X

national digital health platform in the future, in order to support more widespread and structured efforts for early detection and prevention of eye diseases.

DISCUSSIONS

The results of the model test using the Senile Cataract dataset showed that the CNN-based cataract early detection system had excellent performance with an overall accuracy rate of 95%. This value confirms that the model is able to classify in three categories—Immature, Mature, and Normal—with a low error rate. However, while the overall performance was very satisfactory, the test results revealed some important findings that needed further analysis.

First, based on the classification report, the Immature class showed a very high precision value (0.99), but a relatively lower recall (0.87). This difference indicates that although the model is almost always correct when predicting an image as Immature, there are still quite a few Immature images that are not detected and are instead classified into other classes. This indicates that the visual characteristics of the Immature class are similar to the Mature or Normal classes, so some features are not optimally captured by the model. This error is also reinforced by the results of the confusion matrix, which shows that out of 165 Immature images, as many as 13 were incorrectly predicted as Mature and 8 as Normal (). This misidentification can be caused by variations in lens turbidity that are at the threshold between Immature and Mature.

Second, the Mature class showed the highest recall among the cataract classes (0.98), indicating that almost all Mature images were successfully detected. The high detection success can be attributed to the visual characteristics of the mature cataract being clearer and more contrasting, making it easier to recognize by CNN. Although the precision of this class is slightly lower (0.92), the F1-score performance is still very good (0.95). The most common error comes from the Mature image being misidentified as Normal, although there are only three cases. This indicates that certain images may have lighting or shooting angles that make the cloudiness appear less insignificant.

Third, the Normal class showed the most stable performance with a precision of 0.94 and the highest overall recall (0.99). These values show that the model is very reliable in distinguishing normal eyes from eyes with cataracts. There are only two misclassifications, one Normal image each incorrectly classified as Immature and Mature (). These findings suggest that the visual features of normal lenses are much easier to separate than features in the other two classes of cataracts.

Overall, an average precision, recall and F1-score of 0.95 indicates that the model performs consistently across categories. The most dominant misclassification occurs between the Immature and Mature classes, as seen in the confusion matrix. The similarity of visual features such as a less extreme level of opacity makes these two classes more difficult to distinguish. These findings suggest that improving image resolution, adding training data, or applying additional feature engineering techniques has the potential to improve performance in difficult classrooms.

In the implementation of web-based systems, test results show that the integration of the CNN model into the application is going well. Users can upload eye images and receive diagnostic results in real time. Based on the interface illustration in the result document (), the application is designed to be simple, responsive, and easy to understand. The presence of this web-based system makes cataract detection technology more accessible to the wider community, including areas with limited health facilities. In addition, data security aspects such as encryption and privacy protection increase the reliability of the system for use at scale.

Overall, the results of the study show that the CNN model developed has achieved excellent performance and has great potential to be implemented in digital service-based cataract early detection systems. However, some aspects such as the differentiation between Immature and Mature classes and the variation in image quality can still be improved in future research. This improvement effort can be done by enlarging the number of datasets, fine-tuning the CNN architecture more deeply, and applying advanced augmentation techniques. Thus, this system can be an effective solution to support automated cataract diagnosis on web-based health platforms in the future.

CONCLUSION

Based on the results of the research and evaluation that has been carried out, an automatic cataract detection system has been successfully developed using the Convolutional Neural Network (CNN) method with VGG-16 architecture through a transfer learning approach. The "Senile Cataract" dataset from the Kaggle platform was used to train and test the model in classifying eye images into three categories, namely Normal, Immature Cataract, and Mature Cataract. The system stages include image pre-processing, feature extraction, model training, and performance evaluation using accuracy, precision, recall, f1-score, and confusion matrix metrics. The test results show that the model is capable of achieving 95% accuracy, with the highest f1-score in the Normal class at 0.96. Although there is misclassification between the Immature and Mature classes, the overall performance of the model is considered excellent and consistent in each class.

*name of corresponding author

The accuracy of the VGG16 model, which reached 95.15%, showed very competitive performance. This result exceeds the accuracy achieved by Chakraborty & Tharini (2020), which only reached 90% in cataract binary classification. In addition, this accuracy is also higher than the research by Ganokratanaa et al. (2023), which applied CNN and obtained an accuracy of 93.5% in two-class classification. This success can be attributed to an effective fine-tuning strategy and the addition of a dropout layer that reduces overfitting. However, this accuracy is still slightly below, which reached 96.2%, but that study used a pre-processed dataset from a standardized clinical source.

For further research, it is recommended to conduct prospective testing (clinical trials) with datasets collected directly from hospitals and verified by medical specialists to assess the model's performance in real clinical conditions, explore and compare the performance of lighter CNN architectures to reduce model complexity in web and mobile deployment, and develop the model into a mobile application (iOS/Android) to improve portability and accessibility for people in remote areas.

With these achievements, the developed cataract detection system has great potential to be implemented in web-based health services as a fast, accurate, and easily accessible early diagnosis tool, especially in areas with limited access to medical personnel.

REFERENCES

Aghababaeyan, Z., Abdellatif, M., Briand, L., Ramesh, S., & Bagherzadeh, M. (2023). Black-Box Testing of Deep Neural Networks through Test Case Diversity. *IEEE Transactions on Software Engineering*, *49*(5), 3182–3204. https://doi.org/10.1109/TSE.2023.3243522

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data 2021 8:1*, *8*(1), 53-. https://doi.org/10.1186/S40537-021-00444-8

Andreas, E., Widhiarso, W., Kunci, K., & Mata Katarak;, P. (2023). Klasifikasi Penyakit Mata Katarak Menggunakan Convolutional Neural Network Dengan Arsitektur Inception V3. *MDP Student Conference*, *2*(1), 107–113. https://doi.org/10.35957/MDP-SC.V2I1.3660

Arkin, E., Yadikar, N., Xu, X., Aysa, A., & Ubul, K. (2022). A survey: object detection methods from CNN to transformer. *Multimedia Tools and Applications 2022 82:14*, *82*(14), 21353–21383. https://doi.org/10.1007/S11042-022-13801-3

Chakraborty, P., & Tharini, C. (2020). Pneumonia and Eye Disease Detection using Convolutional Neural Networks. *Engineering, Technology & Applied Science Research*, *10*(3), 5769–5774. https://doi.org/10.48084/ETASR.3503

Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. Bin, Islam, K. R., Khan, M. S., Iqbal, A., Emadi, N. Al, Reaz, M. B. I., & Islam, M. T. (2020). Can AI Help in Screening Viral and COVID-19 Pneumonia? *IEEE Access*, *8*, 132665–132676. https://doi.org/10.1109/ACCESS.2020.3010287

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature 2017 542:7639*, *542*(7639), 115–118. https://doi.org/10.1038/nature21056

Firdaus, D. H., Imran, B., Bakti, L. D., & Suryadi, E. (2022). KLASIFIKASI PENYAKIT KATARAK BERDASARKAN CITRA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS WEB. *Jurnal Kecerdasan Buatan Dan Teknologi Informasi*, *1*(3), 18–26. https://doi.org/10.69916/JKBTI.V1I3.6

Flaxman, S. R., Bourne, R. R. A., Resnikoff, S., Ackland, P., Braithwaite, T., Cicinelli, M. V., Das, A., Jonas, J. B., Keeffe, J., Kempen, J., Leasher, J., Limburg, H., Naidoo, K., Pesudovs, K., Silvester, A., Stevens, G. A., Tahhan, N., Wong, T., Taylor, H., … Zheng, Y. (2017). Global causes of blindness and distance vision impairment 1990–2020: a systematic review and meta-analysis. *The Lancet Global Health*, *5*(12), e1221–e1234. https://doi.org/10.1016/S2214-109X(17)30393-5

Ganokratanaa, T., Ketcham, M., & Pramkeaw, P. (2023). Advancements in Cataract Detection: The Systematic Development of LeNet-Convolutional Neural Network Models. *Journal of Imaging 2023, Vol. 9, Page 197*, *9*(10), 197. https://doi.org/10.3390/JIMAGING9100197

Ghimire, D. (2020). *Comparative study on Python web frameworks: Flask and Django*. http://www.theseus.fi/handle/10024/339796

Khan, M. S. M., Ahmed, M., Rasel, R. Z., & Khan, M. M. (2021). Cataract Detection Using Convolutional Neural Network with VGG-19 Model. *2021 IEEE World AI IoT Congress, AIIoT 2021*, 209–212. https://doi.org/10.1109/AIIOT52608.2021.9454244

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image*

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15642
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Analysis, *42*, 60–88. https://doi.org/10.1016/J.MEDIA.2017.07.005

Matsoukas, C., Haslum, J. F., Sorkhei, M., Söderberg, M., & Smith, K. (2022). *What Makes Transfer Learning Work for Medical Images: Feature Reuse & Other Factors* (pp. 9225–9234).

Nurona Cahya, F., Hardi, N., Riana, D., Hadianti, S., Mandiri Jakarta Cipinang Melayu, N., Makasar, K., Jakarta Timur, K., & Khusus Ibukota Jakarta, D. (2021). Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN). *Sistemasi: Jurnal Sistem Informasi*, *10*(3), 618–626. https://doi.org/10.32520/STMSI.V10I3.1248

Rafli, H., Zen, R., Nuryasin, I., Informatika, P., Malang, U. M., Raya, J., No, T., 246, K., & Lowokwaru, K. M. (2024). PENERAPAN WHITEBOX TESTING PADA PENGUJIAN SISTEM MENGGUNAKAN TEKNIK BASIS PATH. *JOISIE (Journal Of Information Systems And Informatics Engineering)*, *8*(1), 101–111. https://doi.org/10.35145/JOISIE.V8I1.4229

Resnikoff, S., Felch, W., Gauthier, T. M., & Spivey, B. (2012). The number of ophthalmologists in practice and training worldwide: a growing gap despite more than 200 000 practitioners. *British Journal of Ophthalmology*, *96*(6), 783–787. https://doi.org/10.1136/BJOPHTHALMOL-2011-301378

Richard, M. A., Paul, C., Nijsten, T., Gisondi, P., Salavastru, C., Taieb, C., Trakatelli, M., Puig, L., & Stratigos, A. (2022). Prevalence of most common skin diseases in Europe: a population-based study. *Journal of the European Academy of Dermatology and Venereology*, *36*(7), 1088–1096. https://doi.org/10.1111/JDV.18050

Rismayani, Pineng, M., & Herlinda. (2022). Using Artificial Neural Network for System Education Eye Disease Recognition Web-Based. *Journal of Biomimetics, Biomaterials and Biomedical Engineering*, *55*, 262–274. https://doi.org/10.4028/P-7Z9XPT

Shidik, F. A., Musthofa, K., Kartiningtyas, A. P., & Agustin, T. (2024). Analisis Citra Medis untuk Identifikasi Penyakit Mata dengan Teknologi Convolutional Neural Networks. *Prosiding Seminar Nasional Amikom Surakarta*, *2*, 68–80. https://ojs.amikomsolo.ac.id/index.php/semnasa/article/view/648

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/abs/1409.1556v6

Simonyan, K., & Zisserman, A. (2015). *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*. http://www.robots.ox.ac.uk/

Steinmetz, J. D., Bourne, R. R. A., Briant, P. S., Flaxman, S., Taylor, H. R., Jonas, J. B., Abdoli, A., Abrha, W. A., Abualhasan, A., Abu-Gharbieh, E., Adal, T. G., Afshin, A., Ahmadieh, H., Alemayehu, W., Alemzadeh, S. A., Alfaar, A. S., Alipour, V., Androudi, S., Arabloo, J., … Vos, T. (2021). Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to VISION 2020: The Right to Sight: An analysis for the Global Burden of Disease Study. *The Lancet Global Health*, *9*(2), e144–e160. https://doi.org/10.1016/S2214-109X(20)30489-7

Talo, M., Yildirim, O., Baloglu, U. B., Aydin, G., & Acharya, U. R. (2019). Convolutional neural networks for multi-class brain disease detection using MRI images. *Computerized Medical Imaging and Graphics*, *78*, 101673. https://doi.org/10.1016/J.COMPMEDIMAG.2019.101673

Tan, M., & Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks* (pp. 6105–6114). PMLR. https://proceedings.mlr.press/v97/tan19a.html

Ting, D. S. W., Pasquale, L. R., Peng, L., Campbell, J. P., Lee, A. Y., Raman, R., Tan, G. S. W., Schmetterer, L., Keane, P. A., & Wong, T. Y. (2019). Artificial intelligence and deep learning in ophthalmology. *British Journal of Ophthalmology*, *103*(2), 167–175. https://doi.org/10.1136/BJOPHTHALMOL-2018-313173

V, V., K, C. R., & C., R. A. (2022). *Real Time Object Detection System with YOLO and CNN Models: A Review*. https://arxiv.org/pdf/2208.00773

*name of corresponding author