Cataract Classification Project Report

1. Introduction

The goal of this project was to develop a binary classification model that can accurately identify cataracts in eye images. The final model was deployed as an API, allowing users to upload an image (or a single-page PDF), which the model then classifies as either `cataract` or `normal`. This report summarizes the approach taken, the methodology followed, and the results obtained.

---

2. Approach

Model Selection

Two primary models were considered for this classification task:

1. **Vision Transformer (ViT)** embeddings with a non-trainable feature extractor and a custom linear classifier.
2. **ResNet**-based model with non-trainable ResNet embeddings and a custom linear classifier.

After experimentation, the **ViT model** was selected as the optimal architecture due to its superior performance in this task. Using **ViT embeddings (non-trainable) followed by a linear classifier** yielded the highest accuracy, with minimal overfitting observed. ResNet, on the other hand, showed lower accuracy, likely due to the fact that without fine-tuning the feature extractor, it was less effective at generalizing for this task.

Final Model Architecture

The best-performing model was structured as follows:

- **ViT Embeddings (Non-Trainable)**: Pre-trained Vision Transformer feature extractor, with weights frozen to leverage pre-trained embeddings.
- **Classifier Head**:
    - Fully Connected Layer: Maps ViT embeddings to a hidden layer of size 512.
    - ReLU Activation: Applied for non-linearity.
    - Fully Connected Output Layer: Outputs a single value for binary classification.
    - Sigmoid Activation: Applied to obtain a probability score for the binary class.

Data Augmentation

To improve generalization, data augmentation techniques were applied. These included:

- **Random Horizontal Flip**: To simulate slight variations in eye orientation.
- **Rotation**: Images were randomly rotated within a 15-degree range.
- **Resize:** Images were resized to dimension (224,224)
- **Normalization:** Images were normalized to make them similar to VIT pretraining normalization

Data augmentation was effective in enhancing model robustness, particularly given the relatively small dataset size. (Results before and after augmentation mentioned below)

---

3. Methodology

3.1 Data Preprocessing

The images were resized to 224x224 pixels to match the input requirements of ViT. Normalization was applied based on ImageNet statistics to standardize the data across batches.

3.2 Model Development

The model development pipeline followed these key steps:

1. **Feature Extraction with ViT**:
    - ViT embeddings were extracted from a non-trainable pre-trained Vision Transformer model.
    - This setup allowed leveraging rich, high-level features without the need for large computational resources to fine-tune the ViT.
2. **Classifier Head Training**:
    - The custom linear classifier head was trained on the extracted features.
    - Binary Cross-Entropy Loss was used as the loss function, and the optimizer was Adam with a learning rate of 1e-4.
    - Training was conducted over 30 epochs, with performance monitored using accuracy, precision, recall, and F1-score metrics.

3.3 Evaluation

The model was evaluated on the test set, and the following metrics were analyzed:

- **Confusion Matrix**: Provided a breakdown of true positives, false positives, true negatives, and false negatives.
- **ROC Curve and AUC Score**: Visualized the model's performance across different classification thresholds.
- **Classification Report**: Included precision, recall, and F1-score for each class.

These evaluations were conducted using the `evaluate.py` script, which saved performance metrics and generated plots.

3.4 Deployment

The model was deployed as an API using FastAPI, allowing for simple HTTP-based predictions. The API accepts image and single-page PDF uploads, returning a classification (`cataract` or `normal`) and a confidence score. This interface was chosen to make the model accessible to end-users, enabling them to classify images through a simple upload.

---

4. Results

Model Performance

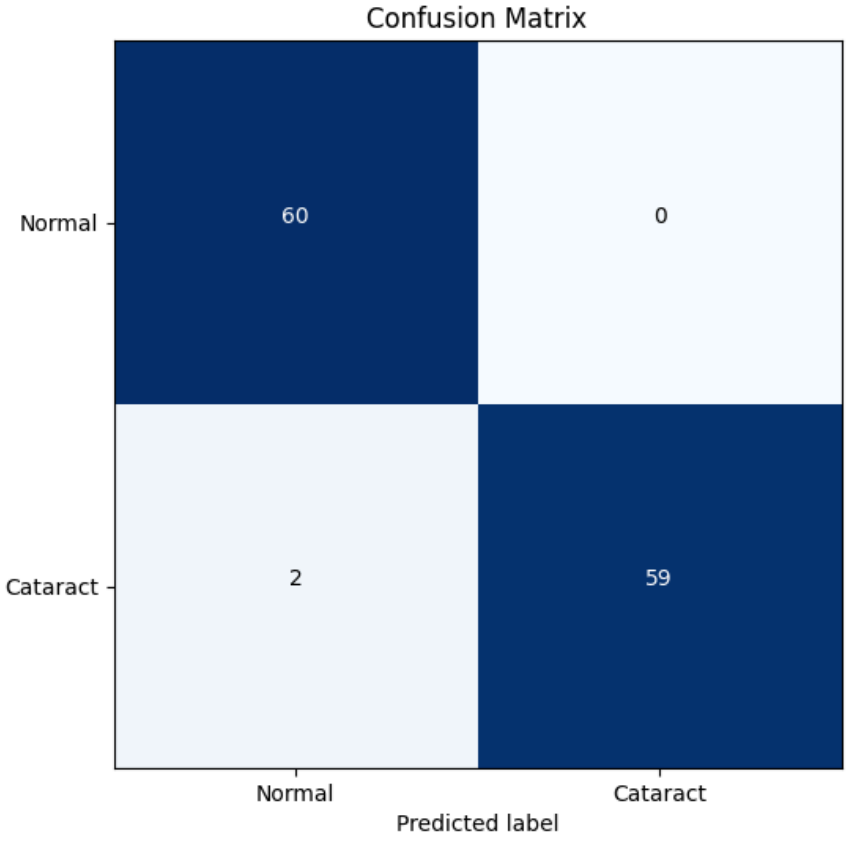
The final ViT-based model achieved the following performance metrics on the test set:

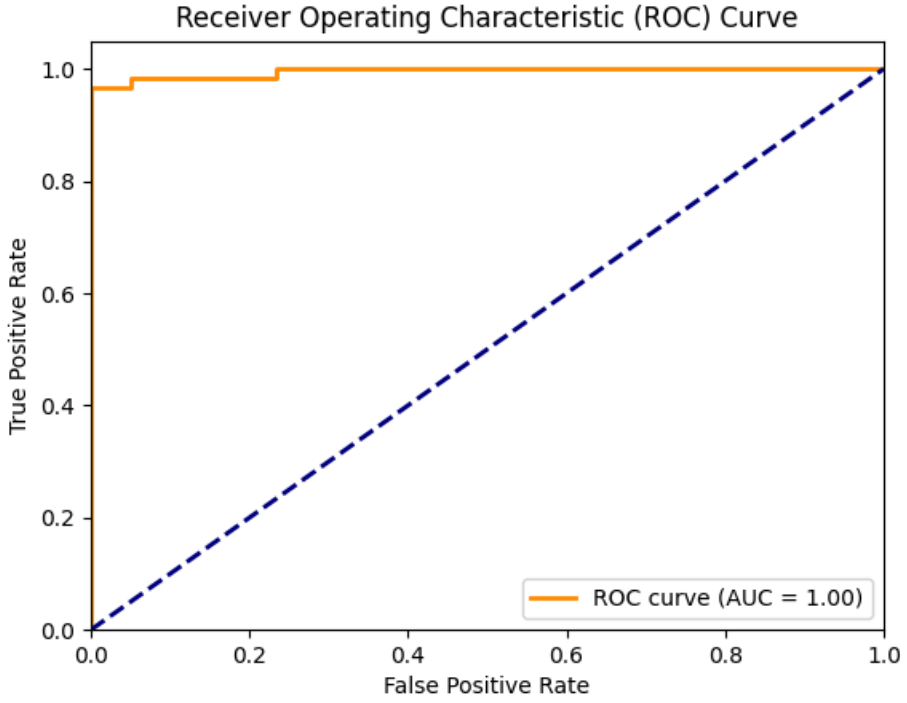
```
Classification Report:
              precision    recall  f1-score   support

      Normal       0.97      1.00      0.98        60
    Cataract       1.00      0.97      0.98        61

    accuracy                           0.98       121
   macro avg       0.98      0.98      0.98       121
weighted avg       0.98      0.98      0.98       121
```

Accuracy: 98.35%

- **AUC Score**: 1.00

Plot attached below


Confusion Matrix
Normal
Cataract
60
0
2
59
Normal
Cataract
Predicted label


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (AUC = 1.00)

Results before Horizontal Flip and Random Rotation Vs After

Horizontal flip was added to adjust and augment left and right eyes in the dataset. Random rotation was added to make the model robust against any variations in clicking pictures. The F1 scores increased from 0.96 to 0.98 after making the discussed changes

	F1 score
Before added Augmentation	**0.96**
After added Augmentations	**0.98**

Comparison with ResNet

Using a non-trainable ResNet for feature extraction was also attempted. However, this approach yielded lower accuracy than the ViT-based approach, as shown below:

Metric	**ViT-based Model**	**ResNet-based Model**
Accuracy	0.98	0.96
F1-Score	0.98	0.96

ResNet's performance was lower, likely due to the model's reduced ability to generalize without fine-tuning. In comparison, the ViT embeddings provided higher-level features, making them more suitable for binary classification without additional training.

---

5. Conclusion

The ViT-based approach proved to be the most effective model for the cataract classification task, achieving high accuracy and robustness with frozen embeddings. By using data augmentation and a lightweight classifier head, the model achieved a balance between simplicity and performance, suitable for deployment as an API.

Key Takeaways

1. **ViT Embeddings**: Pre-trained ViT embeddings outperformed ResNet embeddings in this context, especially with non-trainable weights.
2. **Data Augmentation**: Improved model robustness by simulating real-world variations.
3. **API Accessibility**: FastAPI-based deployment allowed for seamless model access, enabling easy integration with other applications or tools.

Challenges and Future Improvements

Future improvements could include experimenting with fine-tuning the ViT model if computational resources allow, potentially boosting performance further. Additionally, testing on a more diverse dataset could provide insights into model generalization for different eye conditions or image quality variations.
