NephroNet-VGG16: AI-Powered Kidney Disease Classification and Financial Modeling

-Rania Binth Zubair

ABSTRACT

NephroNet-VGG16 is an AI-powered diagnostic tool designed to classify kidney diseases from CT scan images using a fine-tuned VGG16 convolutional neural network. Leveraging the power of transfer learning, the model is pre-trained on ImageNet and fine-tuned for kidney disease classification, providing high accuracy and reliability. This report outlines the technical development, market potential, and financial viability of the product. With a user-friendly interface and integration capabilities, NephroNet-VGG16 has the potential to revolutionize kidney disease diagnosis in medical institutions worldwide, enhancing diagnostic efficiency and patient care.

INTRODUCTION

Chronic kidney disease (CKD) is a widespread health issue affecting millions globally, leading to significant morbidity and mortality. Early diagnosis is crucial for effective treatment, yet traditional diagnostic methods are often resource-intensive and rely on specialized expertise, leading to potential delays. With advancements in artificial intelligence (AI) and deep learning, automated medical diagnostics are becoming more accessible and efficient. NephroNet-VGG16, an AI-powered tool that utilizes the VGG16 convolutional neural network model, aims to classify kidney diseases from CT scans with high accuracy, offering a faster, more reliable diagnostic process. This report examines the feasibility, market potential, and financial viability of NephroNet-VGG16 as a commercial healthcare product.

PROBLEM STATEMENT

Kidney disease, particularly chronic kidney disease (CKD), is challenging to diagnose early due to subtle symptoms and the complexity of analyzing CT scans, which heavily rely on specialist expertise. The increasing global burden of CKD, combined with limited access to nephrologists in many regions, has created a diagnostic bottleneck, leading to delays in treatment and inconsistent results. Current diagnostic tools either lack specialization for kidney disease or are not widely accessible. NephroNet-VGG16 seeks to address this gap by providing an AI-driven solution that accurately classifies kidney diseases from CT scans, improving diagnostic efficiency and accessibility.

Prototype Selection

Feasibility

- The model uses VGG16 pre-trained on ImageNet, which is widely accepted in medical imaging and has proven efficiency in disease classification tasks.
- Development Timeline: 2-3 years for a fully functional product with potential regulatory approvals and clinical validation.
- Required Infrastructure: High-performance computing resources for model training, a secure cloud platform, and access to labeled kidney CT scans.

Viability:

- Long-term Use: Chronic kidney disease (CKD) is a global health issue, with rising cases due to aging populations and lifestyle changes. The relevance of AI in healthcare diagnostics is expected to grow.
- Adaptability: The system can evolve to include other types of kidney diseases or imaging modalities (e.g., MRI, ultrasound), ensuring it remains useful over the long term (20-30 years).

Monetization

- Direct: Monetize through a subscription-based model for healthcare institutions, pay-per-use diagnostics, or licensing the technology to diagnostic centers.
- Indirect: Avoid models like free access with only research-driven revenue (not applicable here). Focus on monetization directly tied to diagnostics or image analysis.

Prototype Development

Data training

```
class Training:
    def __init__(self, config: TrainingConfig):
        self.config = config


    def get_base_model(self):
        self.model = tf.keras.models.load_model(
            self.config.updated_base_model_path
        )

    def train_valid_generator(self):

        datagenerator_kwargs = dict(
            rescale = 1./255,
            validation_split=0.20
        )

        dataflow_kwargs = dict(
            target_size=self.config.params_image_size[:-1],
            batch_size=self.config.params_batch_size,
            interpolation="bilinear"
        )

        valid_datagenerator = tf.keras.preprocessing.image.ImageDataGenerator(
            **datagenerator_kwargs
        )

        self.valid_generator = valid_datagenerator.flow_from_directory(
            directory=self.config.training_data,
            subset="validation",
            shuffle=False,
```

Data configuration

In [8]:

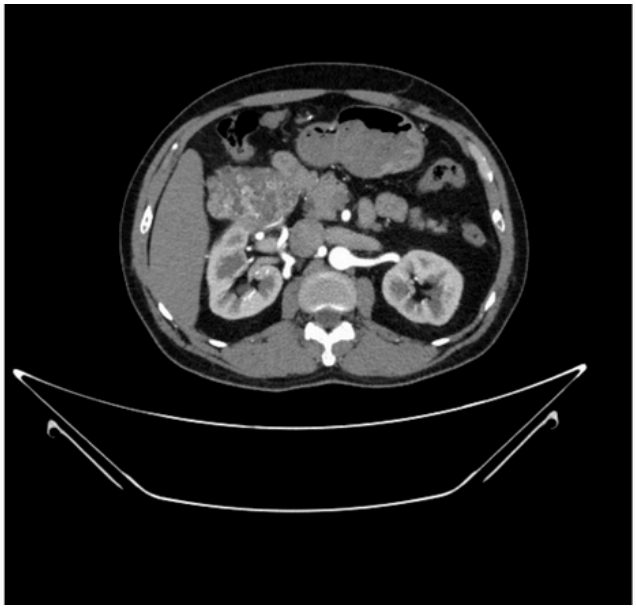
```
class ConfigurationManager:
    def __init__(
        self,
        config_filepath = CONFIG_FILE_PATH,
        params_filepath = PARAMS_FILE_PATH):

        self.config = read_yaml(config_filepath)
        self.params = read_yaml(params_filepath)

        create_directories([self.config.artifacts_root])



    def get_training_config(self) -> TrainingConfig:
        training = self.config.training
        prepare_base_model = self.config.prepare_base_model
        params = self.params
        training_data = os.path.join(self.config.data_ingestion.unzip_dir, "kidney-ct-scan-image")
        create_directories([
            Path(training.root_dir)
        ])

        training_config = TrainingConfig(
            root_dir=Path(training.root_dir),
            trained_model_path=Path(training.trained_model_path),
            updated_base_model_path=Path(prepare_base_model.updated_base_model_path),
            training_data=Path(training_data),
            params_epochs=params.EPOCHS,
            params_batch_size=params.BATCH_SIZE,
            params_is_augmentation=params.AUGMENTATION,
            params_image_size=params.IMAGE_SIZE
```

Data Evaluation:

In [15]:

```
class Evaluation:
    def __init__(self, config: EvaluationConfig):
        self.config = config


    def _valid_generator(self):

        datagenerator_kwargs = dict(
            rescale = 1./255,
            validation_split=0.30
        )

        dataflow_kwargs = dict(
            target_size=self.config.params_image_size[:-1],
            batch_size=self.config.params_batch_size,
            interpolation="bilinear"
        )

        valid_datagenerator = tf.keras.preprocessing.image.ImageDataGenerator(
            **datagenerator_kwargs
        )

        self.valid_generator = valid_datagenerator.flow_from_directory(
            directory=self.config.training_data,
            subset="validation",
            shuffle=False,
            **dataflow_kwargs
        )
```

Testing on an image:



Business Model Development

This section will outline the commercial viability, business strategies, and market positioning of NephroNet-VGG16.

Target Audience:

- Healthcare Providers: Hospitals, clinics, and diagnostic centers.
- Medical Research Institutions: Research groups focused on nephrology and medical imaging.
- Government and Non-Profit Health Organizations: National health agencies interested in improving CKD diagnostics.

Revenue Streams:

- Software Licensing: Hospitals and diagnostic centers can license NephroNet-VGG16 on a yearly basis.
- Pay-per-use Diagnostic Tool: Pay-per-scan model for smaller clinics.
- Partnerships with Medical Device Manufacturers: Integrate NephroNet-VGG16 into existing radiology equipment.

Business Scalability:

- Geographic Expansion: Roll out the product in stages, starting from major metropolitan cities to rural areas.
- Partnership with Telemedicine Providers: Collaborate with remote healthcare services for kidney diagnostics.

Competitive Landscape:

- Analyze competitors in the AI healthcare market, particularly those working in diagnostic imaging.
- Highlight the competitive advantages of NephroNet-VGG16: higher accuracy, ease of use, pre-trained model, and integration potential with existing hospital workflows.

Financial modelling

Financial Equation

Given the growing demand for AI in healthcare diagnostics, particularly in disease detection, we can use a linear growth model to forecast the financials of NephroNet-VGG16. The equation will follow the same structure as your example:

$$\text{Profit}(y) = \text{Revenue from Sales}(mx) - \text{Costs}(c)$$

Where:

- **Revenue from Sales (mx)** = Price per Service × Number of Scans Processed
- **Costs (c)** = Fixed Costs + Variable Costs

Assumptions:

- **Price per service (Price per scan)**: $50 (cost per scan charged to hospitals or diagnostic centers)
- **Number of scans processed**: 10,000 (scans per year)
- **Fixed costs**: $50,000 (for infrastructure, software development, regulatory compliance, etc.)
- **Variable costs**: $20,000 (server usage, maintenance, customer support, marketing, etc.)

Calculations:

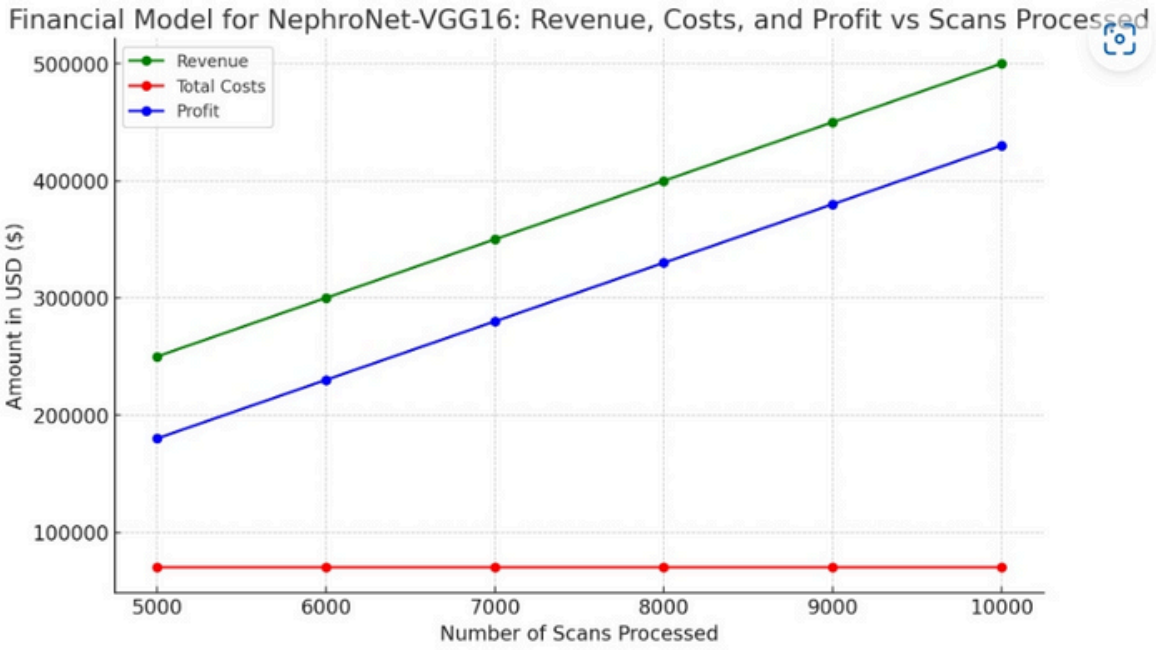
- **Revenue from Sales (mx)** = $50 × 10,000 = $500,000
- **Costs (c)** = $50,000 + $20,000 = $70,000

Therefore, **Profit**:

$$\text{Profit} = 500,000 - 70,000 = \$430,000$$

Conclusion:

Based on these assumptions, the NephroNet-VGG16 project could generate a profit of $430,000 per year, provided the market for kidney disease diagnostics continues to grow, and the service processes 10,000 scans annually. This model helps forecast profitability and aids in planning for future scaling and market expansion.



Here is a graph representing the **Revenue**, **Total Costs**, and **Profit** for NephroNet-VGG16 as the number of scans processed increases. The graph shows the financial trends based on the provided financial model, highlighting the relationship between the number of scans processed and the potential profitability of the project.

GIthub link:

https://github.com/RaniaBZ/Feynn_labs-final_project