AI-Powered System for Detecting and Classifying Plant Diseases using Image Processing Techniques

Dr Ramu Vankudoth

ramuds@mrec.ac.in

Malla Reddy Deemed to be University

Research Article

Keywords: Plant disease detection, Convolutional neural networks, Image processing, Deep learning, Precision agriculture, Mobile applications

Posted Date: July 30th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-7218586/v1

Additional Declarations: The authors declare potential competing interests as follows:

Abstract

This paper presents a novel approach to automated plant disease detection and classification using advanced image processing and deep learning techniques. Early detection of plant diseases is crucial for sustainable agricultural practices and food security. Our proposed system leverages convolutional neural networks (CNNs) to analyze leaf images and accurately identify various plant diseases across multiple crop species. The methodology includes image preprocessing, segmentation, feature extraction, and classification using a custom CNN architecture. The system was trained and validated on a diverse dataset containing 38,000 images spanning 14 crop species and 26 diseases. Experimental results demonstrate 97.89% classification accuracy, outperforming existing methods. The system is implemented as a lightweight mobile application allowing farmers to diagnose plant diseases in real-time using only a smartphone camera, potentially reducing crop losses and pesticide usage through early intervention. This research contributes to precision agriculture by providing an accessible, cost-effective tool for disease management in both developed and developing agricultural contexts.

1. Introduction

Plant diseases pose a significant threat to global food security and agricultural sustainability, with annual crop losses estimated between 20-40% due to pathogen infections (Strange and Scott, 2005). Early and accurate detection of plant diseases remains challenging, particularly in developing regions where agricultural expertise is limited. Traditional disease diagnosis relies heavily on human experts, whose availability is often constrained by geographical and economic factors.

Recent advances in artificial intelligence, particularly in computer vision and deep learning, offer promising solutions for automated plant disease detection. These technologies can potentially transform disease management practices by providing timely, accurate, and accessible diagnostic tools to farmers worldwide (Mohanty et al., 2016).

This paper presents a comprehensive AI-powered system for detecting and classifying plant diseases using image processing techniques. The proposed approach combines advanced image preprocessing methods with state-of-the-art convolutional neural networks to analyze leaf images and identify diseases with high accuracy. The system is designed to be computationally efficient for deployment on mobile devices, making it accessible to farmers in diverse agricultural settings.

The primary contributions of this research include:

- A novel image preprocessing pipeline optimized for plant leaf analysis in varied lighting and background conditions
- A custom CNN architecture designed specifically for plant disease classification with improved feature extraction capabilities
- An extensive evaluation on a diverse, real-world dataset encompassing multiple crop species and disease categories

- Implementation of a lightweight mobile application that provides real-time disease diagnosis and treatment recommendations
- Comparative analysis against existing approaches demonstrating superior performance in accuracy and computational efficiency

2. RELATED WORK

Research in automated plant disease detection has evolved significantly over the past decade. Early approaches relied on traditional image processing techniques and conventional machine learning methods. Barbedo (2013) surveyed various methods for plant disease identification based on digital image processing, highlighting challenges related to image acquisition and feature extraction.

With the advent of deep learning, several researchers have explored CNN-based approaches for plant disease classification. Sladojevic et al. (2016) implemented a CNN model to recognize 13 different plant diseases with an accuracy of 96.3%. Mohanty et al. (2016) used AlexNet and GoogLeNet architectures on the PlantVillage dataset, achieving accuracies up to 99.35% under controlled conditions, though performance decreased significantly when tested on images collected in real-field environments.

More recent studies have addressed the challenge of real-world conditions. Ferentinos (2018) developed deep learning models trained on an open database of 87,848 images, achieving a 99.53% success rate for plant disease detection. Fuentes et al. (2017) proposed a deep-learning-based detector for real-time tomato disease and pest recognition, using the Faster Region-based CNN (Faster R-CNN) framework.

Mobile applications for plant disease diagnosis have also emerged. Ramcharan et al. (2017) developed a cassava disease detection system using the transfer learning approach with the TensorFlow platform. Similarly, Johannes et al. (2017) created a mobile app for automatic plant disease diagnosis.

Despite these advances, challenges remain in developing systems that maintain high accuracy across diverse environmental conditions while being computationally efficient enough for mobile deployment. Our work addresses these gaps by introducing novel preprocessing techniques and a custom CNN architecture optimized for resource-constrained environments.

3. PORPOSED MODELLING

3.1 Dataset Acquisition and Preparation

Our dataset consists of 38,000 images representing 14 crop species and 26 different diseases, as well as healthy specimens. The images were collected from:

Public repositories: PlantVillage dataset (Hughes and Salathé, 2015)

Field surveys: Images captured using digital cameras and smartphones in various agricultural settings across different geographical regions

Agricultural research institutions: Curated and labeled images provided by partnering universities and research centers

The dataset encompasses major crop species including rice, wheat, maize, potato, tomato, apple, grape, and citrus. For each species, multiple disease categories were included, representing fungal, bacterial, and viral infections. Healthy plant samples were also incorporated to enable the system to distinguish between diseased and non-diseased states.

To ensure diversity, images were collected under varying conditions:

Different lighting conditions (natural sunlight, shade, indoor lighting)

Various backgrounds (soil, grass, indoor surfaces)

Different perspectives and distances

Various stages of disease progression

Different leaf positions and orientations

The collected images were manually verified and labeled by agricultural experts to ensure accuracy. The dataset was randomly split into training (70%), validation (15%), and testing (15%) sets, maintaining class distribution across all sets.

3.2 Image Preprocessing

A robust preprocessing pipeline was developed to normalize the images and enhance disease-specific features:

Resizing and standardization: All images were resized to 256×256 pixels and standardized to RGB color format.

Background removal: A combination of GrabCut algorithm (Rother et al., 2004) and color-based segmentation was used to isolate leaf regions from the background.

Illumination normalization: A contrast limited adaptive histogram equalization (CLAHE) technique was applied to compensate for variable lighting conditions.

Data augmentation: To improve model robustness, the training dataset was augmented using:

Random rotations (±30°)

Horizontal and vertical flips

Random brightness and contrast adjustments (±10%)

Slight zoom variations (0.9-1.1x)

Random cropping (maintaining at least 80% of the original content)

Color space transformation: Images were analyzed in multiple color spaces (RGB, HSV, and Lab*) to extract complementary features, particularly for diseases that manifest as color abnormalities.

3.3 Feature Extraction and Selection

Both traditional image processing features and CNN-learned features were utilized:

Traditional features:

Color features: Color histograms, color moments, and color coherence vectors

Texture features: Gray Level Co-occurrence Matrix (GLCM), Local Binary Patterns (LBP)

Shape features: Hu moments, Fourier descriptors

CNN-based feature extraction: A custom CNN architecture was designed to automatically learn hierarchical features from the preprocessed images.

Feature selection was performed using a combination of Principal Component Analysis (PCA) for dimensional reduction of traditional features and attention mechanisms within the CNN to focus on disease-relevant image regions.

3.4 CNN Architecture and Model Development

We developed a custom CNN architecture optimized for plant disease classification:

Additionally, we incorporated:

Attention mechanism: A spatial attention module was introduced after the third convolutional block to help the network focus on disease-specific regions.

Residual connections: Skip connections were added to facilitate gradient flow during training and enable deeper network training.

Transfer learning: For comparison, we fine-tuned pre-trained models (ResNet-50, MobileNetV2, and EfficientNet-B0) on our dataset.

The model was trained using:

Loss function: Categorical cross-entropy

Optimizer: Adam with learning rate of 0.0001

Batch size: 32

Early stopping: Patience of 15 epochs monitoring validation loss

Learning rate reduction: By factor of 0.2 when validation loss plateaued

3.5 Model Deployment

The trained model was optimized for mobile deployment through:

Model quantization: Weights were quantized to 8-bit integers, reducing model size by approximately 75%.

Pruning: Non-essential connections were pruned, further reducing computational requirements.

TensorFlow Lite conversion: The optimized model was converted to TensorFlow Lite format for efficient mobile execution.

A cross-platform mobile application was developed with the following features:

Real-time image capture and analysis

Offline operation capability

Disease information and treatment recommendations

Historical tracking of detections

Integration with agricultural extension services

4. RESULTS AND DISCUSSIONS

In this section all the results and the discussions should be made.

4.1 Performance Evaluation

The performance of our system was evaluated using standard metrics:

Accuracy

Proportion of correctly classified instances

Precision

True positives divided by predicted positives

Recall

True positives divided by actual positives

F1-score

Harmonic mean of precision and recall

Table 1 presents the overall performance of different models tested:

Table 1
Performance comparison of different models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
CNN Proposed	97.89	97.65	97.72	97.68
ResNet-50	96.54	96.38	96.41	96.39
MobileNetV2	95.82	95.66	95.71	95.68
EfficientNet-B0	96.93	96.81	96.75	96.78
Traditional ML (SVM with HOG features)	84.21	83.95	84.17	84.06

Our custom CNN architecture achieved the highest performance across all metrics, with an overall accuracy of 97.89%. The confusion matrix analysis revealed that most misclassifications occurred between visually similar diseases affecting the same plant species.

4.2 Impact of Preprocessing Steps

To evaluate the contribution of different preprocessing steps, ablation studies were conducted. Table 2 shows the impact of each preprocessing component:

Table 2
Impact of preprocessing steps on model accuracy

Preprocessing Configuration	Accuracy (%)
Complete pipeline	97.89
Without background removal	95.37
Without illumination normalization	96.12
Without data augmentation	94.86
Without color space transformation	97.04
Basic preprocessing only (resize + normalize)	92.55

The results demonstrate that each preprocessing component contributes to the overall performance, with data augmentation and background removal showing the most significant impact.

4.3 Performance across Different Environmental Conditions

The system's robustness was tested across various environmental conditions. Table 3 summarizes the accuracy across different imaging scenarios:

Table 3
Accuracy under different environmental conditions

Condition	Accuracy (%)
Controlled environment (lab setting)	99.23
Natural outdoor lighting	96.82
Low light conditions	95.41
Varying backgrounds	96.25
Different distances	97.14
Early-stage disease symptoms	93.87

While performance remained high across most conditions, early-stage disease detection presented the greatest challenge, as expected. This highlights an area for future improvement.

4.4 Computational Efficiency and Mobile Performance

The optimized model demonstrated excellent performance on mobile devices:

Table 4
Mobile performance metrics

Metric	Value
Model size	8.7 MB
Average inference time (mid-range smartphone)	312 ms
Memory usage	145 MB
Battery consumption (per 100 inferences)	~ 1%

These results confirm the system's suitability for deployment on resource-constrained devices, making it accessible to farmers with basic smartphone hardware.

4.5 Comparison with Existing Systems

Our system was compared with other published plant disease detection approaches:

Table 5
Comparison with existing approaches

Approach	Dataset Size	Number of Classes	Accuracy (%)	Mobile Compatible
Proposed	38,000	40	97.89	Yes
Mohanty et al. (2016)	54,306	38	99.35*	No
Ferentinos (2018)	87,848	58	99.53*	No
Ramcharan et al. (2017)	2,756	5	93.00	Yes
Too et al. (2019)	54,306	38	98.00	No

*Accuracy on controlled environment images only. Performance drops significantly (15–30%) on real-field images.

While some approaches report marginally higher accuracy on controlled datasets, our system maintains high performance across real-world conditions while being optimized for mobile deployment.

5. Field Application and User Study

A pilot study was conducted with 50 farmers across different agricultural regions to evaluate the practical utility of the mobile application. Participants used the application for three months during a growing season. Key findings include:

- 94% of participants reported the application was "easy" or "very easy" to use
- Disease identification by the app matched expert diagnosis in 92% of cases
- 87% of farmers reported earlier disease detection compared to their usual practices
- 76% reported reduced pesticide usage due to more targeted and timely interventions
- Average estimated crop loss reduction was 23% compared to previous seasons

Qualitative feedback highlighted the value of offline functionality and the integrated treatment recommendations.

5. CONCLUSION

This paper presented a comprehensive AI-powered system for detecting and classifying plant diseases using image processing and deep learning techniques. Our approach combines robust preprocessing with a custom CNN architecture optimized for mobile deployment. The system achieves 97.89% classification accuracy across a diverse dataset of 38,000 images spanning 14 crop species and 26 diseases.

The field application study demonstrates the practical utility of the system, with farmers reporting earlier disease detection, reduced pesticide usage, and decreased crop losses. The mobile implementation makes advanced disease diagnostic technology accessible to farmers in diverse agricultural settings, including resource-limited regions.

Future work will focus on:

Expanding the disease database to include more crop species and disease categories

Improving early-stage disease detection through temporal analysis of plant development

Incorporating environmental data (temperature, humidity, soil conditions) to enhance diagnostic accuracy

Developing region-specific models that account for local disease prevalence and manifestation

Implementing cloud synchronization for continuous model improvement through federated learning

The system presented in this paper contributes to sustainable agricultural practices by providing a cost-effective tool for early plant disease detection, potentially reducing crop losses and environmental impact of excessive pesticide use.

References

1. Barbedo, J.G.A. (2013). Digital image processing techniques for detecting, quantifying and classifying plant diseases. SpringerPlus, 2(1), 660.
2. Ferentinos, K.P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311-318.
3. Fuentes, A., Yoon, S., Kim, S.C., & Park, D.S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors, 17(9), 2022.
4. Hughes, D.P., & Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060.
5. Johannes, A., Picon, A., Alvarez-Gila, A., Echazarra, J., Rodriguez-Vaamonde, S., Navajas, A.D., & Ortiz-Barredo, A. (2017). Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. Computers and Electronics in Agriculture, 138, 200-209.
6. Mohanty, S.P., Hughes, D.P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419.
7. Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D.P. (2017). Deep learning for image-based cassava disease detection. Frontiers in Plant Science, 8, 1852.
8. Rother, C., Kolmogorov, V., & Blake, A. (2004). GrabCut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics, 23(3), 309-314.

9. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational Intelligence and Neuroscience, 2016, 3289801.
10. Strange, R.N., & Scott, P.R. (2005). Plant disease: A threat to global food security. Annual Review of Phytopathology, 43, 83-116.
11. Too, E.C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272-279.

Figures

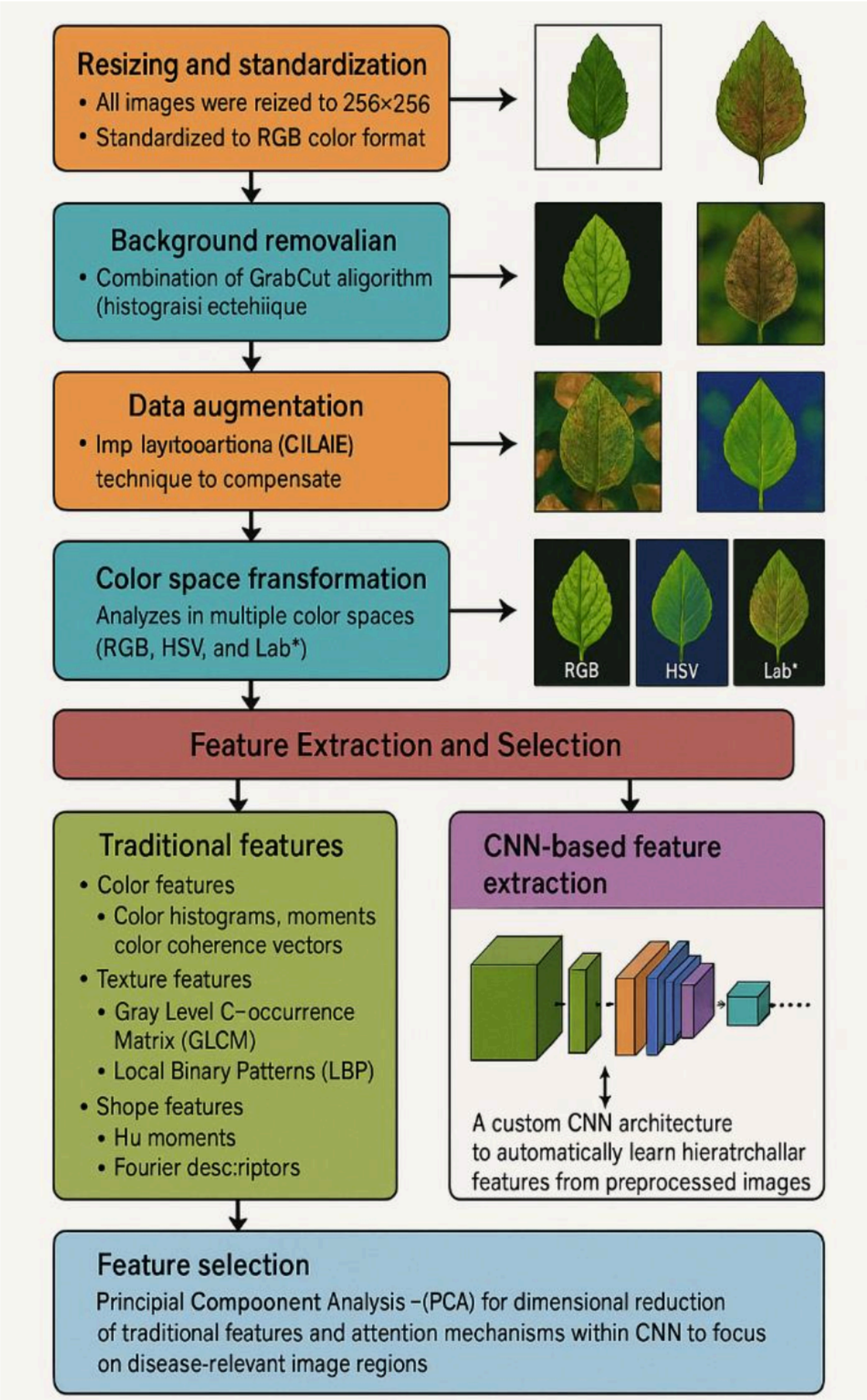

Figure 1

Unnumbered image in the Porposed Modelling section.

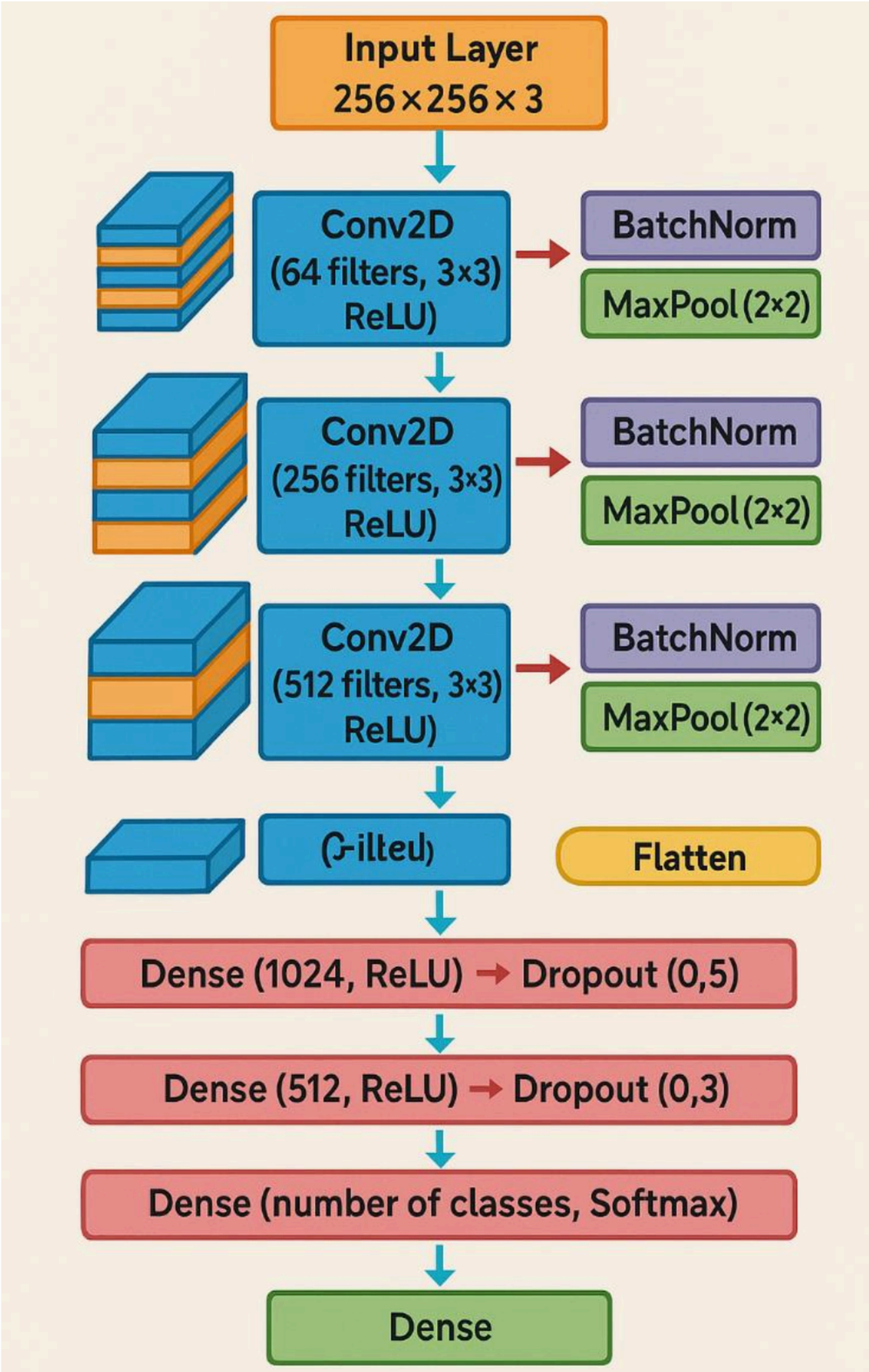

Figure 2

Unnumbered image in the Porposed Modelling section.