**Project Design Phase-II**

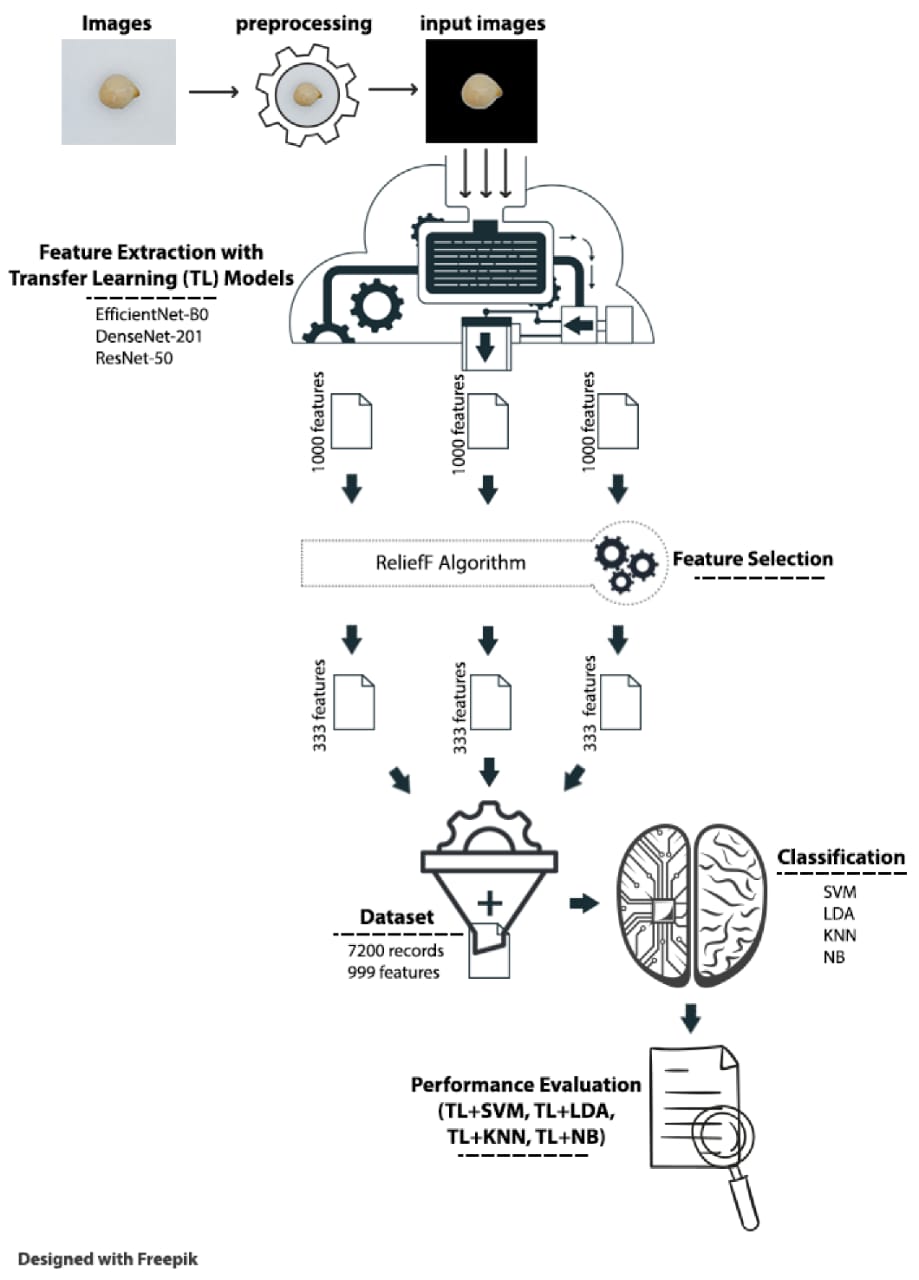
**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 30 June 3035 |
| Team ID | LTVIP2025TMID35759 |
| Project Name | Transfer Learning-Based Classification Of Poultry Diseases For Enhanced Health Management |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The following diagram represents the system architecture of the proposed solution.It include the architectural diagram as below and the information as per the table1 & table 2.

**Example: Poultry Disease Classification Using Transfer Learning Techniques**



## Guidelines:

The infrastructure is divided into **local components** (image capture via mobile/web) and **cloud services** (model hosting, processing, storage, and admin dashboard). It integrates **external interfaces** like Firebase Auth for login, WhatsApp/Twilio API for alerts, and Google Maps API for location tagging. **Data storage** is managed via cloud storage (e.g., AWS S3/Firebase) for images and logs, and cloud databases for user and prediction data. The **machine learning model** (e.g., ResNet50 via TensorFlow/Keras) is deployed in the cloud and accessed through a REST API.

**Table-1 : Components & Technologies**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Interface for uploading poultry images and viewing results | Web UI (HTML, CSS, JavaScript, Bootstrap) |
|  | Application Logic-1 | Image preprocessing and data pipeline | (OpenCV, NumPy, Pandas Python) |
|  | Application Logic-2 | Transfer Learning-based disease classification | TensorFlow/ Keras+ ResNet50 |
|  | Application Logic-3 | Visualization and report generation | Matplotlib, Seaborn, Report Lab |
|  | Database | To store metadata and diagnosis logs | SQLite / MySQL |
|  | Cloud Database | Centralized cloud database for deployment | Firebase Realtime DB / MongoDB Atlas |
|  | File Storage | Storage for training images and test data | Local Filesystem / AWS S3 |
|  | External API-1 | To retrieve disease details / symptoms | Custom Poultry Disease Info API |
|  | External API-2 | Weather data affecting poultry health | Open Weather API |
|  | Machine Learning Model | Poultry disease classification using ResNet50 + fine-tuning | Pre-trained ResNet50 via TensorFlow |
|  | Infrastructure  (Server / Cloud) | Deployment and testing platform | Localhost, Google Colab, AWS EC2 (Cloud) |

**Table-2: Application Characteristics**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Frameworks and libraries used in model and app development | TensorFlow, keras, Flask, OpenCV |
|  | Security Implementations | Secure upload, access control, and encrypted data handling | SHA-256, HTTPS, Firebase Auth |
|  | Scalable Architecture | Microservices-based ML model API, loosely coupled frontend/backend | Docker, REST API |
|  | Availability | Cloud-hosted backend and DB, backups enabled | AWS EC2, Load Balancer (optional) |
|  | Performance | Fast response via preloaded models, image caching, CDN usage | Redis Cache (optional), Flask Caching |