Phase 3 Prototype Development Report: AI-Powered Chronic Disease Risk Screener Using   Non-Invasive Signals

Team Members: Niyati Nikunj Kapadia, Teli Sumit Laxmanbhai, Sarah Pradhan, Patel Jugal Kunteshkumar
Course: CECS 551– Term Project
Date: November 14, 2025

1. **Proof of Concept (PoC)**
    It aimed to validate whether multimodal deep-learning techniques could effectively predict chronic disease risks using non-invasive inputs. This stage was crucial in confirming that combining visual signals (such as nail-bed and retinal images) with tabular health data could yield accurate and interpretable results.

    To achieve this, lightweight yet powerful architectures—**MobileNet V3** and **EfficientNet-B0**—were employed for visual feature extraction, while a **Transformer-based branch** was used to process tabular inputs such as age, blood pressure, and lifestyle indicators. These feature representations were then fused through a custom embedding layer, producing disease-specific probability outputs.

    We integrated **Grad-CAM** for image interpretability, highlighting the areas of visual focus in predictions, and **SHAP** for tabular interpretability, providing transparency in how input features influenced decisions.

    Results:

    - **Nailbed Model:** F1 Score = 0.91
    - **Retina Model:** F1 Score = 0.98
    - **Hypertension Model:** F1 Score = 0.84

2. **Minimum Viable Product (MVP / Prototype)**
    We used Flask Rest API and a React-based web dashboard to build a user-ready prototype. Users can upload non-invasive inputs such as nail-bed or retina images, which are processed through the deep-learning pipeline to generate disease risk probabilities for conditions like Diabetes, Hypertension, and kidney disease. Results are presented as Low, Medium, or High risk, offering clear and immediate feedback as shown in below figure 1&2.

3. **Deep Learning Model Implementation**
    The models were trained using TensorFlow, Scikit-learn and Python 3.10 on a GPU environment, utilizing mixed-precision training to optimize computation and reduce memory usage. Images and tabular features underwent **standard preprocessing**, including normalization and scaling based on distributions identified in Phase 2. To reduce class imbalance, the team implemented **class-weighted loss functions** and **targeted augmentations** such as rotation, flipping, and contrast adjustments. These enhancements improved model robustness, especially in detecting less frequent disease cases.

    The project used a **fixed train–test split** to keep the evaluation process consistent across all experiments. A specific portion of the data was set aside for training, validation, and testing, and these same subsets were used throughout model development. This approach ensured fair performance comparison and made it easier to track improvements during tuning. While it offered less variation than resampling methods, it provided a stable and reproducible setup for all three disease models.

    Overall, this adaptive and iterative training process produced consistent performance across all disease-specific models, maintaining F1-scores above 0.8 and confirming the stability of the multimodal approach.

4. **API Integration & Real-World Applicability**
   The developed deep-learning models were wrapped into a Flask REST API, which interacts seamlessly with the React-based front end. This setup enables fast and scalable communication between the model server and the client interface, ensuring predictions can be generated and displayed in real time.

   In real-world scenarios, the system is designed for **community clinics, telemedicine platforms, and remote healthcare programs**. By enabling low-cost, rapid, and interpretable health screening, the system has the potential to significantly improve **preventive healthcare accessibility** and assist clinicians in early diagnosis—particularly in regions where medical imaging infrastructure is limited.

Conclusion

The prototype phase turned the initial feasibility concept into a working multimodal disease-risk screening system. By combining image and tabular data through deep learning, the model achieved high accuracy and clear interpretability. The web-based MVP, built with React and Flask, delivers real-time predictions with an intuitive interface. Efficient training, explainable outputs, and secure API integration proved the system's reliability on modest hardware. Overall, the AI-powered screener is ready for practical, low-cost healthcare deployment.

Figure 1 & 2 Screenshot of Prototype