CSE 499A: Project Work | Section: 15 | Group: 07 | Members: Amit Chandra Das (2014242042), Tanushree Das (2212225042), Ashraful Islam (2212669042)

Project Progress Report: Arsenic Skin Detection using Machine Learning

I. Overall Summary of Progress

This week, our group's focus shifted from data preparation to the core implementation and training of our two proposed models. This work was divided among the team members, with a primary focus on getting initial performance results. We successfully implemented and trained both our **Baseline CNN** and our first attempt at an **Advanced ResNet50** model. However, both models perfformed significantly below expectations, with the Base line CNN achieving only **63% accuracy** and the ResNet50 model achieving **77% accuracy**. Our primary finding this week is that both initial models are flawed. The baseline is too simple and unstable, while the ResNet50 implementation has a critical bug (likely in the input layer or training configuration) that is preventing it from learning effectively. Our main priority is now to debug and re-build the advanced model correctly.

II. Individual Contributions this Week

A. Amit Chandra Das (Lead Implementation & Model Debugging)

- As the lead implementer, my work this week focused on writing the code for both models, training them, and diagnosing the poor results.
- Baseline CNN Implementation: Wrote, compiled, and trained the simple, from-scratch CNN model.
- Baseline Failure Analysis: The model's test accuracy was a failure at **63%**. The detailed classification report (see Table 1) showed a catastrophic **recall of 39% for 'Infected' cases**. This proves the model is unusable as it misses 61% of all sick patients.
- Advanced ResNet50 Implementation: Wrote the code for our advanced model. This first attempt used UpSampling2D layers to match the ResNet50 input shape, which we now believe was the source of the problem.
- Advanced Model Debugging: This model also failed, achieving only **77% accuracy** far below the 97.69% benchmark. I have concluded that this low score is due to a bug in the implementation. Forcing the model to train on" stretched" (upsampled) images is providing blurry, low-quality data. Furthermore, the model's training configuration (e.g., trainable = False) may not have been set correctly, causing it too overfit.
- My main task for the next update is to delete this buggy ResNet50 code and re-implement it correctly using a frozen base and the proper in put shape=(128,128,3)

B. **Tanushree Das (Data Pipeline Management & Visualization)**

My role this week was to support the modeling process by managing the data pipeline and visualizing the training results.

- Data Pipeline Verification: Ensured that the finalized, scaled datasets (x_train_scaled, y_train, etc.) were correctly formatted and ready to be fed into the model training scripts.

- Training Visualization: Took the history objects from both of Amit's trained models and wrote the Python scripts (using matplotlib) to plot the training/valida tion accuracy and loss graphs.
- Graph Analysis: The resulting plots confirmed the diagnosis. The baseline CNN's graphs showed clear and rapid overfitting. The buggy ResNet50's graphs were erratic and showed that the validation accuracy was not improving, which supports the conclusion that the model was not learning correctly.

C. Ashraful Islam (Benchmark & Report Analysis)

My work supported the team by analyzing the model outputs and documenting our findings in the context of our project goals.

Benchmark Comparison: Compared our two failed results (63%, 77%) against the 97.69% benchmark from the" ArsenicNet" paper. This context proves that our current results are significant failures, not just low scores.

Result Documentation: Created the detailed classi f ication reports for both failed models (see Table 1 for a summary). I specifically analyzed the baseline's 39% recall, which proves it is critically flawed for a medical project.

Report Compilation: Compiled this "Update 3 Report" for the team, summarizing the tasks, the failed results, and Amit's diagnosis of the buggy ResNet50 code.

III. Consolidated Results & Next Steps

This week was a week of critical failure and debugging. We learned that our simple baseline is not powerful enough, and our first attempt at an advanced model was implemented incorrectly.

TABLE I

Model Performance Comparison

Model	Accuracy	Infected Precision	Infected Recall
Baseline CNN	63.0%	74%	39%
ResNet50 (Buggy)	77.1%	73%	86%

Our main priority is to fix these results. Our next steps are:

1) Delete the buggy ResNet50 code (with UpSam pling2D).

2) Re-implement the ResNet50 model correctly, using input shape=(128,128,3) and convolutional base.trainable = False.

3) Train and evaluate this new, correct model, which we are confident will significantly beat both of our failed scores