Diabetic Foot Exam System

Old Dominion University
Fall 2021
Stephanie Trusty
Table of Contents

1. Problem Description
   a. Background
   b. Project Goals
   c. Dataset
   d. Limitations and Constraints
2. Solution and Implementation
   a. Algorithm Flow
   b. Accuracy and Loss
   c. Test Cases
3. Future Considerations
   a. Recommendations for Improvement
4. Conclusion
Problem Description
What is a Diabetic Foot Exam?

A 3-minute professional evaluation to identify a diabetic patient’s risk of foot ulceration. Exam components include:

- Patient history
- Dermatological assessment
- Musculoskeletal assessment
- Neurological assessment
Project Goals

Dermatological Assessment

- Calluses
- Blisters

Musculoskeletal Assessment

- Foot Deformities
  - Clawtoe
  - Hammertoe
  - Bunion
Project Equipment

- Raspberry Pi 3 Model B
- Raspberry Pi Camera Module v2
- Keyboard
- Mouse
Python Libraries and Packages

- PiCamera
  - Controlling the Raspberry Pi Camera Module
- Matplotlib
  - Python plotting library
- Python Imaging Library
  - Open, save, rotate image files
- TensorFlow
  - Software library for machine learning and AI
Image Dataset

- The dataset consists of 420 images placed into three categories.
  - Callus
  - Blister
  - Deformation
- Training images: 336
- Validation images: 84
Limitations and Constraints

- Limited Dataset Size
  - Ideal size: 1,000 images per class
  - Actual size: 140 images per class
- Lack of Diversity in Data
  - Skin tone
  - Foot Placement
  - Aging Skin
- Address the most predominant condition
Solution and Implementation
Implementation Challenges

- TensorFlow version and Raspberry Pi compatibility
- Data uniqueness
- Dataset selection
- Camera position and lighting
Algorithm Flow

1. Capture image of patient's foot
2. Create rotations of the image
3. Load the trained classification model
4. Make classification prediction on all images
5. Compare scores for each image
6. Present the highest score and its classification
Training and Validation Loss

Training Comparison

- Training Loss (start): 2.1089
- Training Loss (end): 0.4783

Validation Comparison

- Validation Loss (start): 0.9912
- Validation Loss (end): 1.5483
Training and Validation Accuracy

Training Comparison:
- Training Accuracy (start): 0.6042
- Training Accuracy (end): 0.8006

Validation Comparison
- Validation Accuracy (start): 0.5000
- Validation Accuracy (end): 0.7619
Loss and Overfitting

Evidence of some overfitting
Case Study: Raspberry Pi Image

Bottom View
- Prediction: Deformation
- Confidence: 84.54%
- Correct classification: Callus

Side View
- Prediction: Deformation
- Confidence: 96.45%
- Correct classification: Callus
Case Study Results: Dermatological

Prediction: Deformation
Confidence: 92.36%
Correct Classification: Callus

Prediction: Callus
Confidence: 80.73%
Correct Classification: Callus
Case Study Results: Musculoskeletal

**Prediction:** Deformation

**Confidence:** 78.41%

**Correct Classification:** Deformation

**Prediction:** Callus

**Confidence:** 99.62%

**Correct Classification:** Deformation
Future Considerations
Recommendations for Improvement

- Build data set with original images
- Verify data for uniqueness
  - Ensure each image is unique
- Separate deformation categories
  - Determine more accurate characteristics for a class
- Address patients with multiple areas of concern
Conclusion

- Inaccurate results with high confidence levels
- Evidence of overfitting
- Larger, more diverse dataset is needed
- Additional study needed to determine impact of lighting and camera position
References

- https://care.diabetesjournals.org/content/31/8/1679
- https://www.ibm.com/cloud/learn/overfitting
- https://www.tensorflow.org/
- https://keras.io/