Classification of Skin Cancer Images using Topological Data Analysis

MTG 7396: Topological Data Analysis – Gainesville, Florida

James Diffenderfer

December 4, 2019

Overview of Presentation

Classification of Skin Cancer Images using Topological Data Analysis

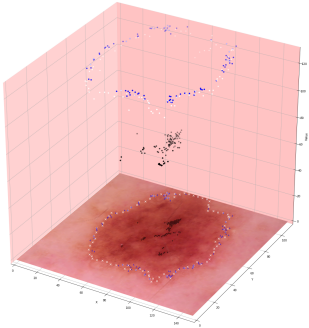
Data Set: Skin Cancer MNIST HAM10000

- Consists of **10015 dermatoscopic images** on various locations of body
- Includes representatives from primary diagnostic categories of pigmented lesions:
  - **MEL: Melanoma**
  - **NV: Melanocytic nevi**
  - AKIEC: Actinic keratoses and Intraepithelial Carcinoma / Bowen's disease
  - BCC: Basal cell carcinoma
  - BKL: Benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses)
  - DF: Dermatofibroma
  - VASC: Vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage)

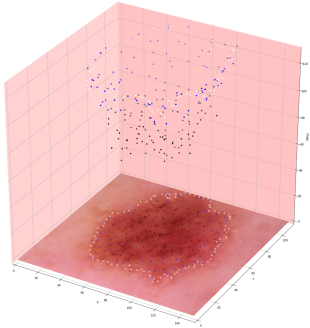
Sampling Points from Skin Cancer Images

- Detected boundary of largest skin cancer region in image
  - Used **cv2.findContours** from the OpenCV library in Python to detect boundary
  - Uniformly sampled 100 points from boundary
- Used thresholding to isolate skin cancer region within image
  - Used **cv2.threshold** from the OpenCV library in Python
- Implemented various methods for sampling pixels after thresholding
  - **Simple**: 100 darkest / 100 lightest pixels using grayscale, RGB, or HSV image format
  - **Cluster**: Cluster pixels by intensity using **sklearn.cluster.KMeans** then sample darkest and lightest points from each cluster using grayscale, RGB, or HSV image format
- **Used 80 images**: Sampled points from 40 images in each class (NV, MEL)

Sampling Points from Skin Cancer Images



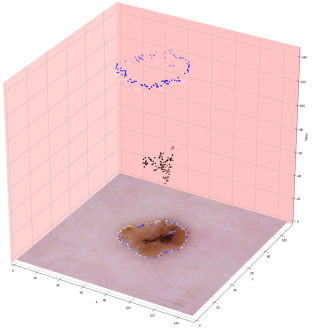
Grayscale Simple Sampling Method



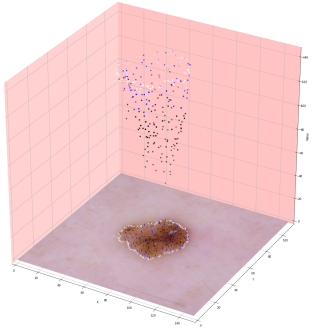
Grayscale Cluster Sampling Method

NV Image: Boundary (white), 100 lightest (blue), and 100 darkest (black) pixels

Sampling Points from Skin Cancer Images



Grayscale Simple Sampling Method



Grayscale Cluster Sampling Method

MEL Image: Boundary (white), 100 lightest (blue), and 100 darkest (black) pixels

Computing Death Vectors and Persistence Landscapes

- Computed persistence landscapes and death vectors with **RStudio TDA package**
  - For each image, we used the sampled 100 boundary points and 100 darkest points from the **simple sampling method** for a total of **200 points per image**
  - Used **Vietoris-Rips complex** in TDA pipeline
- After processing the sampled data in R, for each image we had:
  - One death vector of length 199
  - One persistence landscape vector of length 30100

Principal Component Analysis

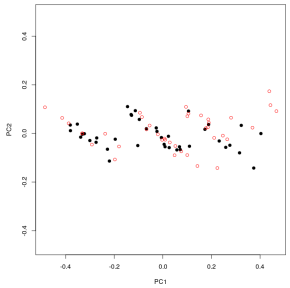


Figure 1: Projection of Death Vectors onto two leading PCA basis vectors

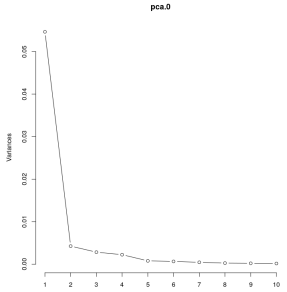


Figure 2: Variance in direction of first ten basis vectors from PCA

Principal Component Analysis

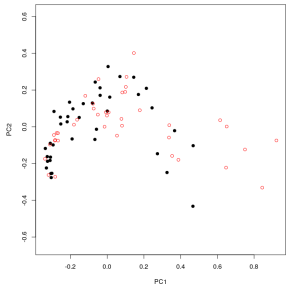


Figure 1: Projection of Pers. Landscapes onto two leading PCA basis vectors

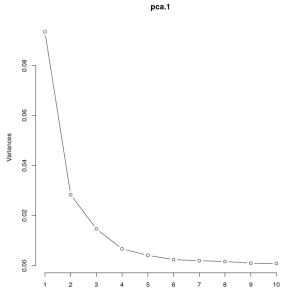
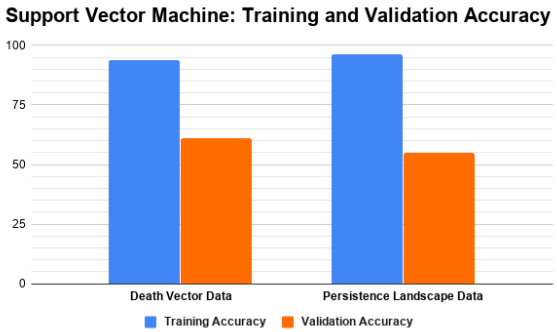


Figure 2: Variance in direction of first ten basis vectors from PCA

Support Vector Machine Classification

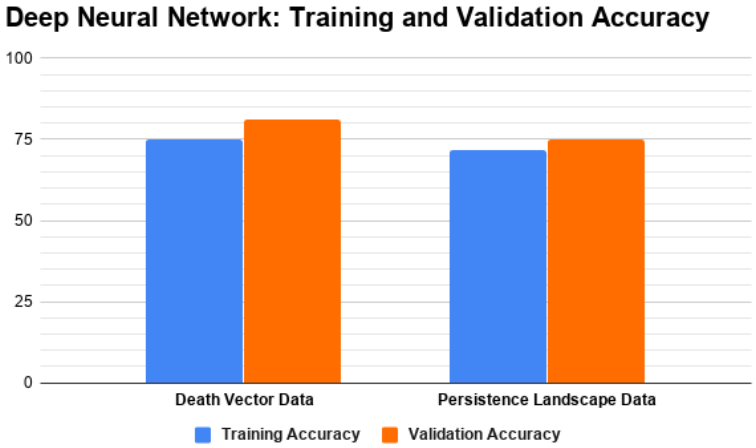


- Used **R package ksvm** with **Gaussian radial basis function kernel**
  - Data does not appear to be lineary separable based on the principal component analysis

Deep Neural Network (DNN) Classification

- Used Python packages **TensorFlow** and **Keras** to model and train DNNs
- Used **Google Colab** with **GPU accelerator** to reduce training time of DNNs
- **Death Vector DNN**:
  - 2 hidden layers, ReLU activation, batch normalization, dropout layers
  - Total of **167,506 trainable parameters**
  - Trained for **120 epochs** using **Adam optimizer** with **batch size of 5**
- **Persistence Landscape DNN**:
  - 1 hidden layer, ReLU activation, batch normalization, dropout layers
  - Total of **3,015,503 trainable parameters**
  - Trained for **150 epochs** using **Adam optimizer** with **batch size of 8**

Deep Neural Network (DNN) Classification



Conclusion and Future Work

- **Successfully modeled and trained classifier for MEL and NV images** that does not suffer from overfitting
- Future work includes:
  - **Update sampling method** to capture slightly more of internal skin cancer structure
  - **Streamline pipeline** so sampling, TDA, and DNN classification can be done in Python
  - **Sample points from more images and more classes** in HAM10000 data set
  - Model and train DNN for **multiclass classification**

Thank you for your attention. Any questions?

UF | UNIVERSITY of FLORIDA