A Supervised Machine Learning Approach for Skin Cancer Detection with Convolutional Neural Networks

Steve Veldman
University of Chicago MS in Applied Data Science
ADSP 31009 Machine Learning and Predictive Analytics – Dr. Arnab Bose, Professor
Submitted 5/22/2024

Github Link: https://github.com/SVeldman/ml_final_spring_2024/tree/main

Background

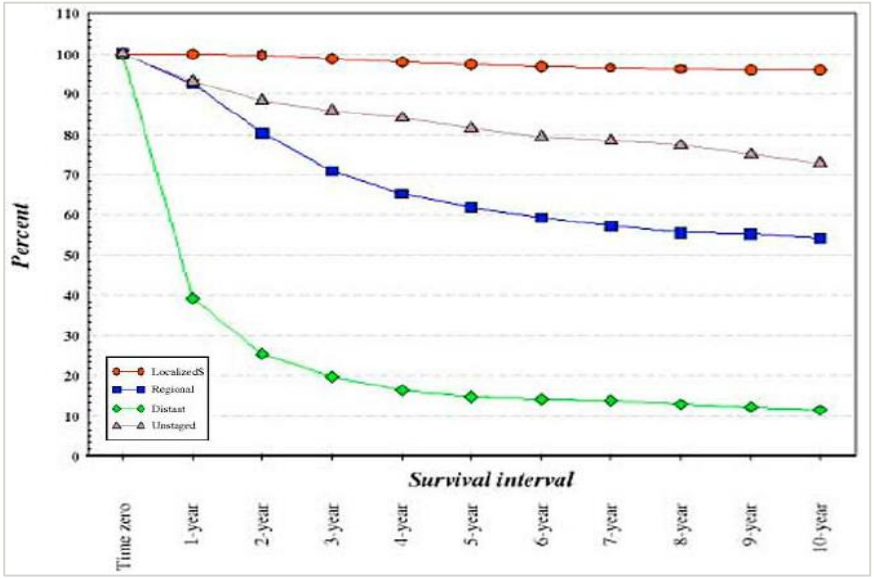
According to the American Academy of Dermatology: *

- Skin cancer is the most common cancer in the United States.
- An estimated 1 in 5 Americans will be diagnosed with a form of skin cancer in their lifetime.
- Nearly 20 Americans die from melanoma (the most common form of skin cancer) every day.
- Early detection greatly increases survival rates among skin cancer patients.

*https://www.aad.org/media/stats-skin-cancer

AI tools have the potential to augment existing diagnostic methods and resources, increasing the number of cases that are identified before they become serious.

Melanoma Survival by Stage at Diagnosis



Source: The Current Epidemiology of Cutaneous Malignant Melanoma (https://pubmed.ncbi.nlm.nih.gov/16368510/)

Problem Statement, Assumptions, and Hypothesis

- The purpose of this project is to develop two Convolutional Neural Networks (CNNs) that can accurately distinguish between the 7 major types of skin lesions based on color image data.
    - Model #1: CNN designed and trained from scratch
    - Model #2: CNN leveraging pretrained model/transfer learning

- Alternatively, this dataset could have been approached as a binary classification problem
    - The multiple classification problem was chosen based on the assumption that it would be more challenging
    - In production, the decision to use binary or multiple-classification would be determined by use-case

- I hypothesize that the transfer learning model will quickly achieve reasonably accurate results, but that with a sufficiently deep/complex architecture and a large enough number of training epochs, the CNN trained from scratch will perform comparably.

Data Overview

- The “HAM10000” dataset consists of 10,015 dermatoscopic images representing the 7 important diagnostic categories of pigmented skin lesions.*
- Dataset was compiled by researched from 4 different sources, including several previously existing images sets.
- Separate test dataset consists of 1,512 additional images.
- Ground Truth for all observations was confirmed via traditional clinical methods (denoted by “dx_type” in metadata).
- Both datasets are publicly available though Harvard Dataverse:
    - https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T

*Tschandl, P., Rosendahl, C. & Kittler, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci Data 5, 180161 (2018). https://doi.org/10.1038/sdata.2018.16

Data Overview

Each dataset contains an associated metadata file that includes:

- Lesion ID and Image ID
- Diagnosis (dx)
- Diagnosis Methodology (dx_type)
- Patient Age
- Patient Sex
- Lesion Location on Body
- Original Dataset (HAM100000 was compiled from 5 sources)

The Seven Categories of Diagnosis:

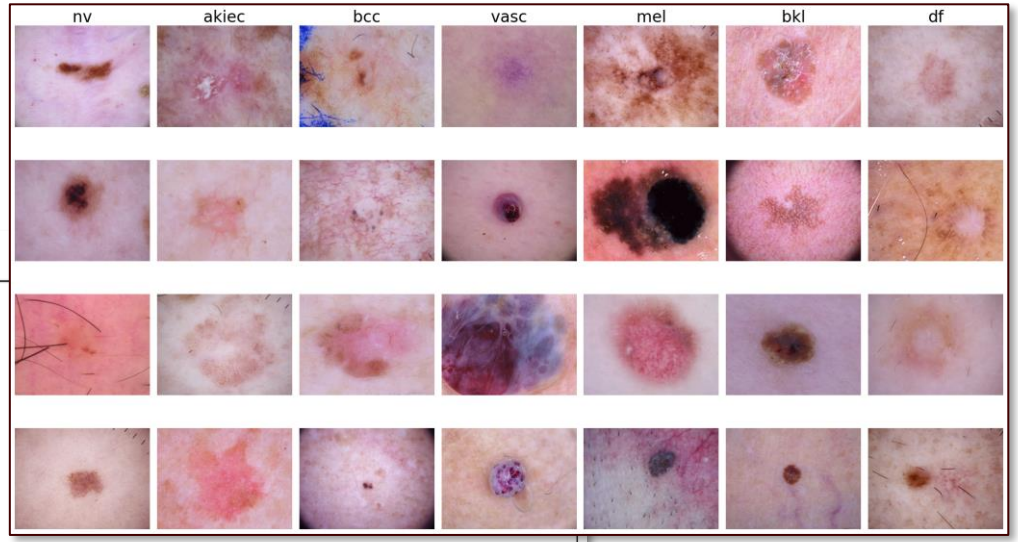
- Actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec)
- Basal cell carcinoma (bcc)
- Benign keratosis-like lesions; solar lentigines / seborrheic keratoses and lichen-planus like keratoses (bkl)
- Dermatofibroma (df)
- Melanoma (mel)
- Melanocytic nevi (nv)
- Vascular lesions; angiomas, angiokeratomas, pyogenic granulomas and hemorrhage (vasc)

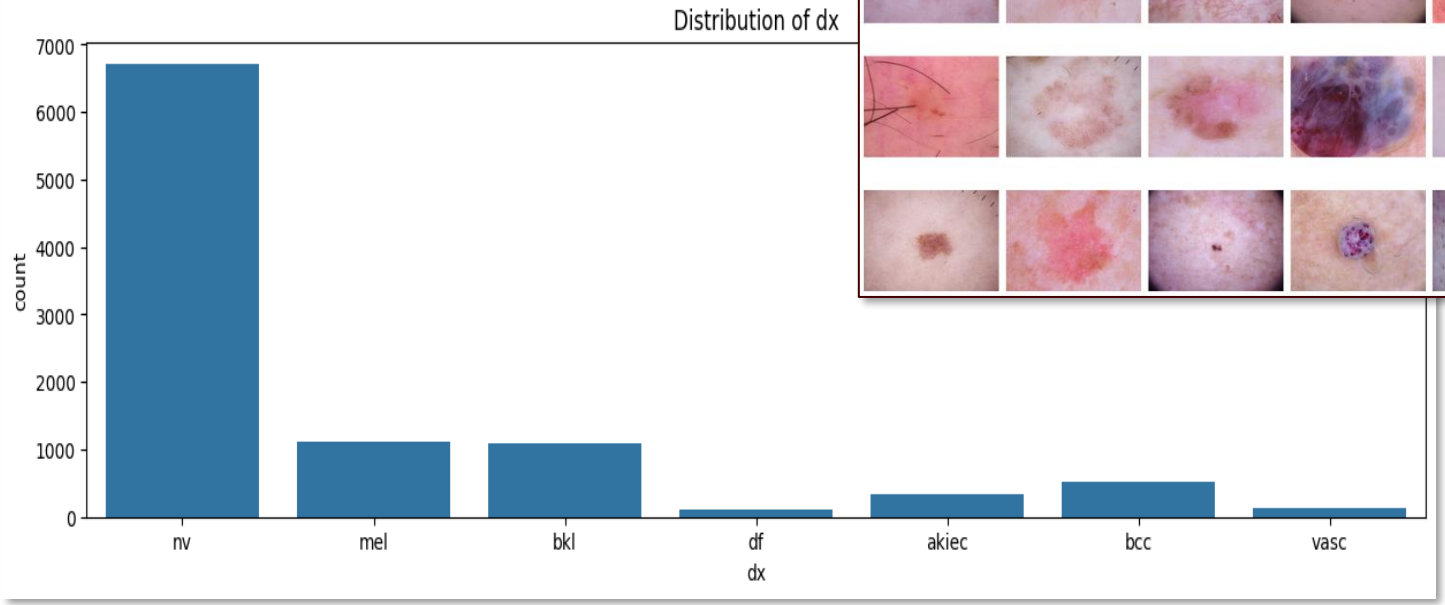
Technical Infrastructure

- Initial “shallow” CNN model was developed in local Jupyter notebook using 1,000 observation sub-sample.
- Due to the size of the full dataset and the computational power required to train full-depth CNN, several cloud-based solutions were explored.
- Ultimately settled on Google Cloud Compute Engine to create a Virtual Machine Instance:
  - 200 GB of cloud storage
  - 1x NVIDIA V100 GPU
  - Instance could be stopped when not in use, but would retain data when restarted
  - Provided greatly increased processing/computing power, while still keeping cost reasonable

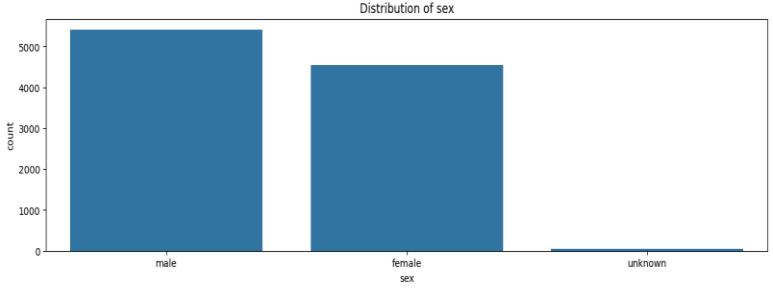
Exploratory Data Analysis

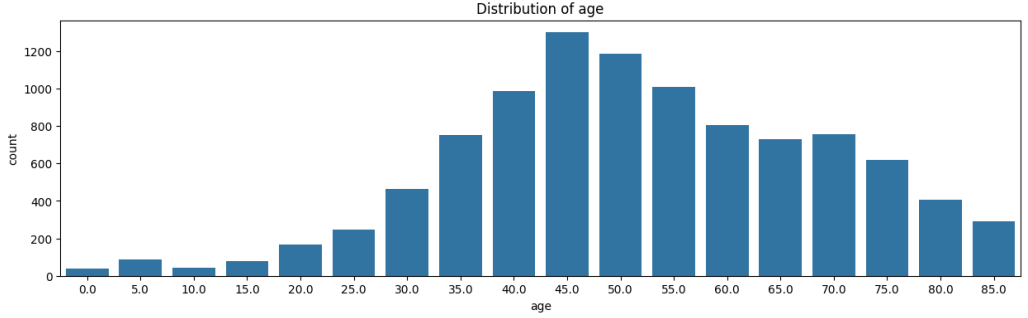
- Classes are heavily imbalanced in favor of melanocytic nevi (nv)
- Visually, each class shows a great deal of variation across a random sampling of images

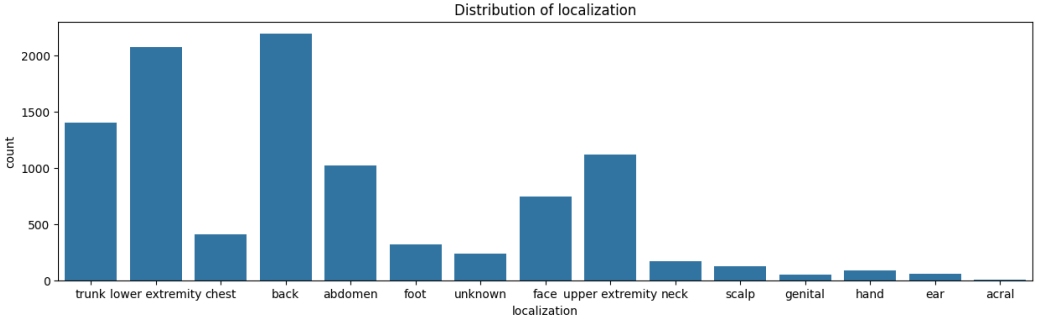


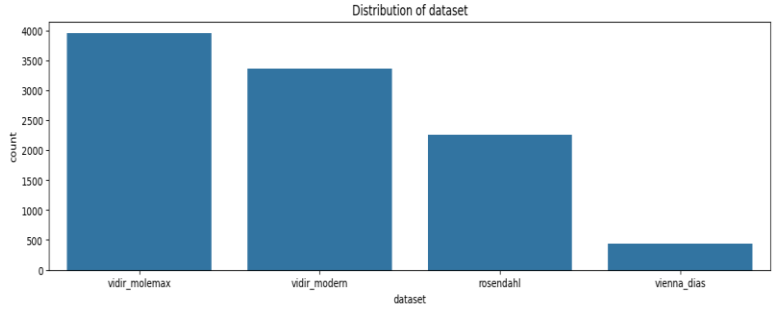


Exploratory Data Analysis - Metadata









Feature Engineering and Transformations (Data Augmentation):

- Image data was organized into folders based on category/label, and imported using TensorFlow's tf.data.Dataset object class as separate training and validation datasets.
- The following transformation layers were built to include with each model:
  - Size and Scale:
    - Resize from 450x600 to 255x300
    - Rescaling (1./255)
  - Data Augmentation:
    - Random Flip
    - Random Rotation
    - Random Zoom
    - Random Contrast

Model Development:

- A great variety of different architectures were explored for both the “from scratch” and transfer learning models (approximately 10+ variation each):
  - “From scratch” model began as simple “shallow” network (input layer, 2 convolutional layers, one dense layer, and output layer) and was gradually expanded until complexity reached the limits of available system resources.
  - Transfer learning models ranged from a single dense layer up to fully developed CNN architecture added after the pretrained ResNet50 model.
- Combinations of the following regularization techniques were explored on each model:
  - Batch Normalization
  - Early Stopping
  - Dropout Layers
  - Learning Rate Adjustments
- Both models were trained using a categorical cross entropy loss function and Adam optimizer.

Model Architectures:

Trained From Scratch:

Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 225, 300, 3)	0
sequential_1 (Sequential)	(None, 225, 300, 3)	0
conv2d_8 (Conv2D)	(None, 225, 300, 128)	3,584
activation_12 (Activation)	(None, 225, 300, 128)	0
batch_normalization_8 (BatchNormalization)	(None, 225, 300, 128)	512
max_pooling2d_5 (MaxPooling2D)	(None, 112, 150, 128)	0
conv2d_9 (Conv2D)	(None, 110, 148, 128)	147,584
activation_13 (Activation)	(None, 110, 148, 128)	0
batch_normalization_9 (BatchNormalization)	(None, 110, 148, 128)	512
max_pooling2d_6 (MaxPooling2D)	(None, 55, 74, 128)	0
conv2d_10 (Conv2D)	(None, 53, 72, 128)	147,584
activation_14 (Activation)	(None, 53, 72, 128)	0
batch_normalization_10 (BatchNormalization)	(None, 53, 72, 128)	512
conv2d_11 (Conv2D)	(None, 51, 70, 128)	147,584
activation_15 (Activation)	(None, 51, 70, 128)	0
batch_normalization_11 (BatchNormalization)	(None, 51, 70, 128)	512
max_pooling2d_7 (MaxPooling2D)	(None, 25, 35, 128)	0
dropout_2 (Dropout)	(None, 25, 35, 128)	0
conv2d_12 (Conv2D)	(None, 23, 33, 128)	147,584
activation_16 (Activation)	(None, 23, 33, 128)	0
batch_normalization_12 (BatchNormalization)	(None, 23, 33, 128)	512
conv2d_13 (Conv2D)	(None, 21, 31, 128)	147,584
activation_17 (Activation)	(None, 21, 31, 128)	0
batch_normalization_13 (BatchNormalization)	(None, 21, 31, 128)	512
conv2d_14 (Conv2D)	(None, 8, 13, 128)	147,584
activation_18 (Activation)	(None, 8, 13, 128)	0
batch_normalization_14 (BatchNormalization)	(None, 8, 13, 128)	512
conv2d_15 (Conv2D)	(None, 6, 11, 128)	147,584
activation_19 (Activation)	(None, 6, 11, 128)	0
batch_normalization_15 (BatchNormalization)	(None, 6, 11, 128)	512
max_pooling2d_9 (MaxPooling2D)	(None, 3, 5, 128)	0
flatten_1 (Flatten)	(None, 1920)	0
activation_20 (Activation)	(None, 1920)	0
dense_3 (Dense)	(None, 128)	245,888
activation_21 (Activation)	(None, 128)	0
dense_4 (Dense)	(None, 128)	16,512
activation_22 (Activation)	(None, 128)	0
dropout_3 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 7)	903
activation_23 (Activation)	(None, 7)	0

Total params: 3,908,119 (14.91 MB)
Trainable params: 1,302,023 (4.97 MB)
Non-trainable params: 2,048 (8.00 KB)
Optimizer params: 2,604,048 (9.93 MB)

Transfer Learning:

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 15, 19, 2048)	23,587,712
global_average_pooling2d_5 (GlobalAveragePooling2D)	(None, 2048)	0
dense_10 (Dense)	(None, 128)	262,272
dropout_5 (Dropout)	(None, 128)	0
dense_11 (Dense)	(None, 7)	903

Total params: 24,377,239 (92.99 MB)
Trainable params: 263,175 (1.00 MB)
Non-trainable params: 23,587,712 (89.98 MB)
Optimizer params: 526,352 (2.01 MB)

Results

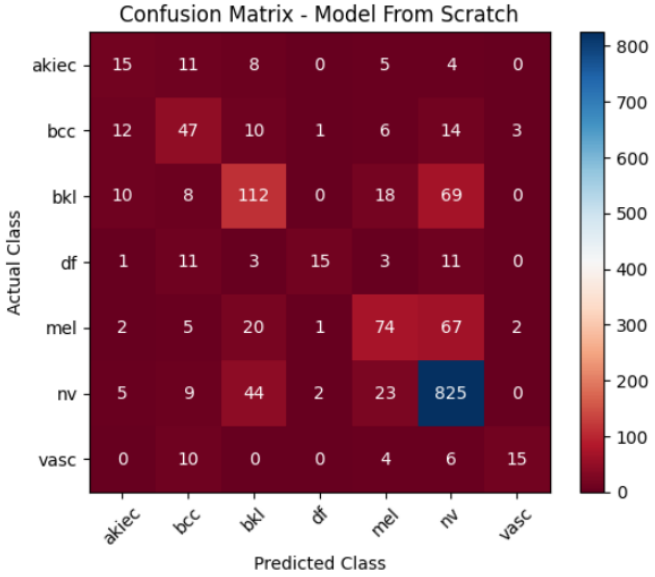
- Model built from scratch benefitted greatly from increasingly complex model architectures and greater number of training epochs. Regularization methods prevented any serious overfitting, with only a modest loss of accuracy from training to validation data and from validation to test data.
- Transfer learning model proved very prone to overfitting, and performed best with minimal layers built on top of the base model and no more than ~10 training epochs.
- Performance of transfer learning model suffered when resizing/rescaling and data augmentation layers were applied (could not test “from scratch” model with full sized images due to system limitations).

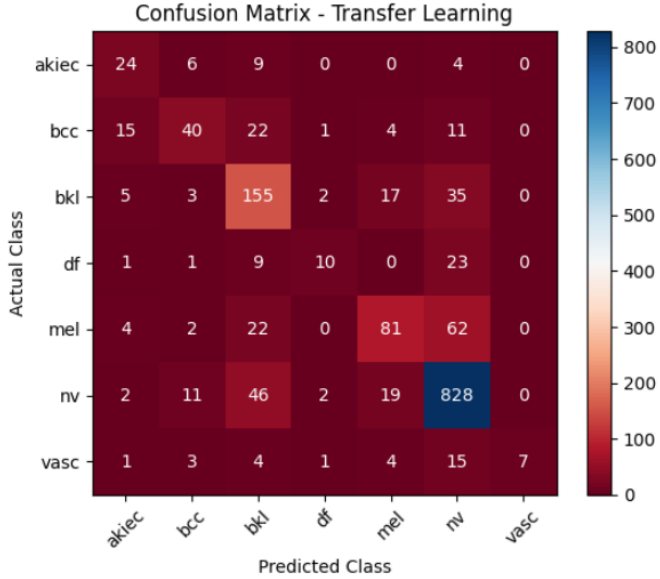
Results

Final models of each type performed realtively similarly across training, validation, and test data:

Model/Accuracy	Training	Validation	Test	# Epochs
From Scratch	85.40%	78.85%	72.17%	40
Pretrained	84.18%	80.47%	76.80%	10

Results





Class Accuracy	akiec	bcc	bkl	df	mel	nv	vasc
From Scratch	34.88%	50.54%	51.61%	34.09%	43.27%	90.86%	42.86%
Pretrained	55.81%	43.01%	71.42%	22.73	47.37%	91.19%	20.00%

Future Work

- Address Data Imbalance:
  - Weighted Loss Functions:
    - Weighted Binary Cross-Entropy – Adjust influence of each class so that training does not focus only on most frequent categories
    - Focal Loss – Incentivizes model to focus training on hard-to-classify examples
  - Adjust Number of Observations:
    - Random Undersampling – Melanocytic nevi represented disproportional amount of dataset)
    - Synthetic Data Generation – Generative Adversarial Network (GAN)
- Additional Models:
  - Binary Classification
  - Incorporate Metadata Information through Hybrid/Blended Model

Conclusions:

- This dataset represents a very challenging classification task
  - As observed in exploratory analysis:
    - Significant variation exists within the examples of each category
    - Visual characteristics that can be shared across multiple categories
  - In practice, even highly trained dermatologists rely on clinical tools such as biopsy to confirm a diagnosis
- Both transfer learning and training a model completely from scratch achieved ~ 75% accuracy predicting over 7 categories
- I high accuracy rate for the majority class ('nv') heavily influences this accuracy rate. After the data imbalance issue is correctly addressed, these models could be valuable tools to aid qualified medical professionals in the identification and diagnosis of skin cancer **- with a proper understanding of their limitations.**

Thank You

Appendix – Proposed Further Improvements to Current Models

- Improve From-Scratch Model:
  - Increase Complexity (would require larger virtual machine):
    - Use full-sized images
    - Larger/deeper model (more layers)
  - Model Tuning:
    - Tune data augmentation layers
    - Alternate loss functions/regularization
- Improve Transfer Learning Model:
  - Unfreeze 2-3 additional layers and retrain
  - Unfreeze whole model and train with very slow learning rate
  - Explore different base/pretrained models

Appendix – Hybrid Model Proposed Architecture

- Hybrid Model with Images and Tabular Data:
  - Build and train CNN with image data as before
    - Generate resultant dense representation
  - Build another neural network based on useful metadata columns:
    - Age
    - Sex
    - Localization
  - Combine both neural networks:
    - Concatenate resultant tensors from both models
    - Pass through softmax/output layer

* Credit: Stack Overflow