ISIC 2019 - Machine learning for Dermatology

Karen Loaiza[1], Kenneth Thomsen[2], and Ole Winther[1]

1 DTU Health Tech, Denmark Technical University ; 2 Dermatology and Venereology, Aarhus University Hospital

Introduction

The International Skin Imaging Collaboration (ISIC) has developed the ISIC Archive, an international repository of dermoscopic images, for both the purposes of clinical training, and for supporting technical research toward automated algorithmic analysis by hosting the ISIC Challenges [1]. The goal for ISIC 2019 is classify dermoscopic images among 9 different diagnostic categories: Melanoma, Melanocytic nevus, Basal cell carcinoma, Actinic keratosis, Benign keratosis (solar lentigo / seborrheic keratosis / lichen planus-like keratosis), Dermatofibroma, Vascular lesion, Squamous cell carcinoma, and none of the others [1].

Key points

- We describe a method for constructing a skin lesion classifier using Python, PyTorch, Skorch, and Scikit-learn for the ISIC 2019 Skin Lesion Classification Challenge with novel state-of-the-art features.

Figure 1: Technologies

- We classified skin lesions based on a dataset of 25,331 dermoscopic images from 8 different classes.
- We created a 3 main component approach with data balancing, progressive image resizing, and Transfer learning.
- Our best model achieves an accuracy of 65% using a stratified 3-fold cross-validation.
- Available at `https://github.com/kalilamali/ISIC_2019_Machine_learning_for_Dermatology`.

Data balancing

Figure 2: Oversampling

Data augmentation

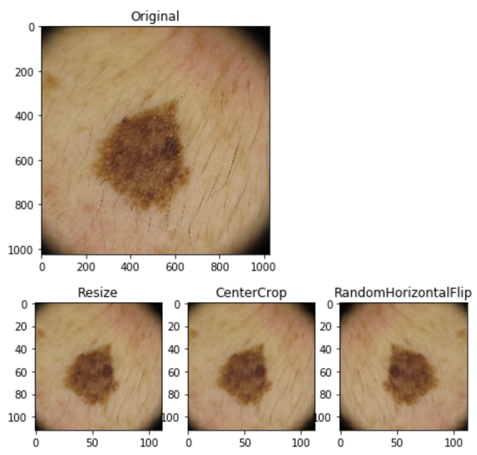


Figure 3: Lesion augmented

Progressive image resizing

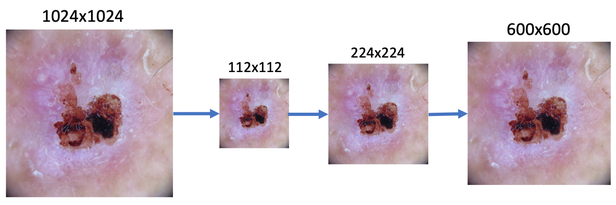


Figure 4: Progressive image resizing

Cross validation

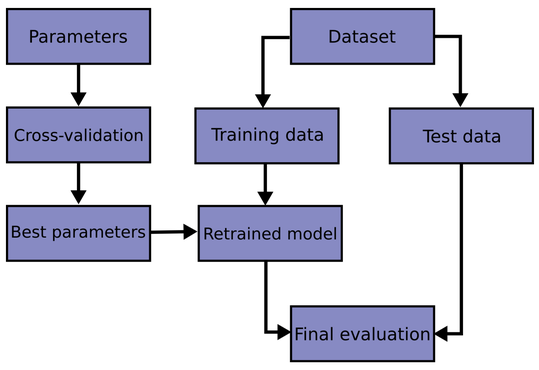


Figure 5: Cross validation workflow [2]

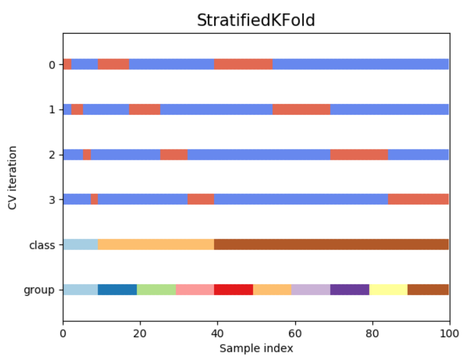


Figure 6: Stratified k-fold [2]

Transfer learning

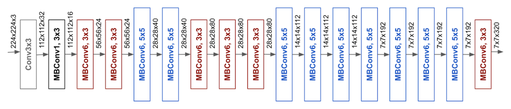


Figure 7: Efficient net B0 baseline [3]

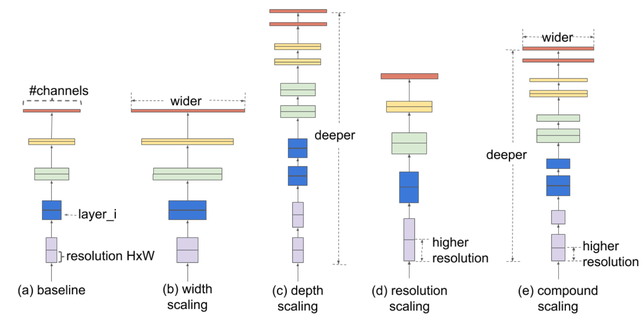


Figure 8: Scaling methods [3]

Modeling

epoch	train_loss	valid_acc	valid_loss	cp	dur
1	1.7120	0.0388	2.8110	+	29.9588
2	1.3966	0.2233	2.1734	+	29.9254
3	1.2133	0.4466	1.6872	+	29.9226
4	1.0703	0.5631	1.5449	+	29.9164
5	0.9871	0.5534	1.4614		29.9115
6	0.9118	0.5728	1.3919	+	29.9041
7	0.8460	0.6019	1.3230	+	29.9254
8	0.8189	0.6117	1.3140	+	29.9190
9	0.7418	0.6019	1.3242		29.9155
10	0.7241	0.6019	1.3144		29.9060
11	0.7215	0.6019	1.3120		29.8829
12	0.7118	0.6019	1.3046		29.8675
13	0.6974	0.6019	1.3131		29.8747
14	0.7214	0.6019	1.3074		29.8787
15	0.6918	0.6117	1.2989		29.8653
16	0.6938	0.6019	1.3017		29.8402
17	0.6978	0.6019	1.2987		29.8272
18	0.7254	0.6019	1.3061		29.8595
19	0.7079	0.6019	1.3068		29.8504
20	0.7180	0.6019	1.3001		29.8551
21	0.6925	0.6019	1.3044		29.8733

Stopping since valid_loss has not improved in the last 5 epochs.

tweet Accuracy reached 0.5 at epoch 4!!! #skorch #pytorch

Figure 9: Net 1 run

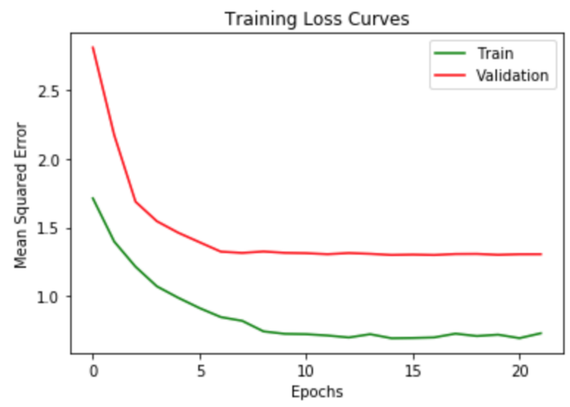
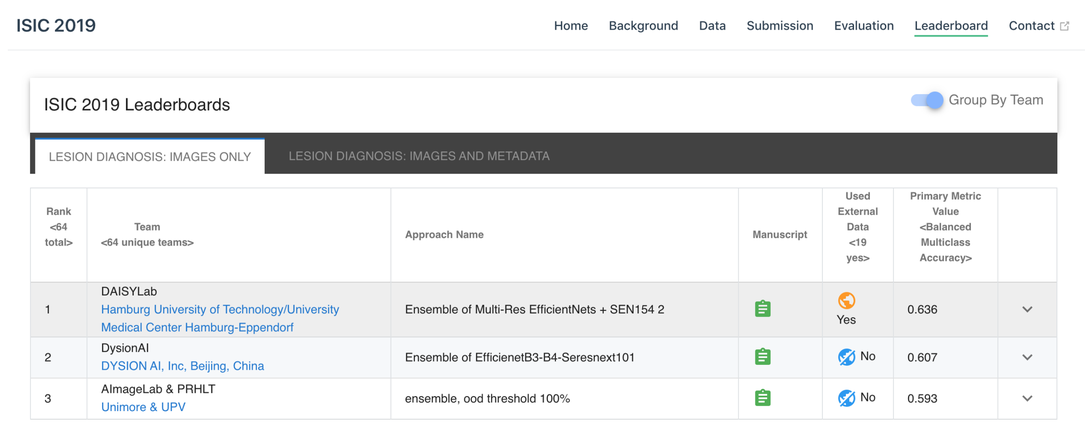


Figure 10: Loss curves

Model performance

Our best model has an accuracy of 65% using a stratified 3-fold cross-validation.



Rank <64 total>	Team <64 unique teams>	Approach Name	Manuscript	Used External Data <19 yes>	Primary Metric Value <Balanced Multiclass Accuracy>	
1	DAISYLab Hamburg University of Technology/University Medical Center Hamburg-Eppendorf	Ensemble of Multi-Res EfficientNets + SEN154 2		Yes	0.636	
2	DysionAI DYSION AI, Inc, Beijing, China	Ensemble of EfficienetB3-B4-Seresnext101		No	0.607	
3	AImageLab & PRHLT Unimore & UPV	ensemble, ood threshold 100%		No	0.593	

Figure 11: ISIC 2019 Leaderboard [1]

Confusion matrices

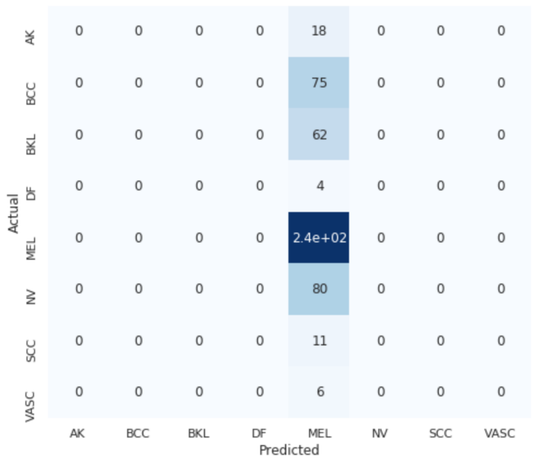


Figure 12: Dummy classifier

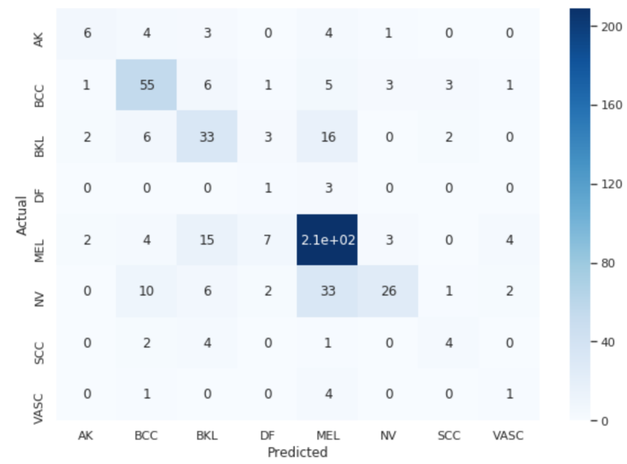


Figure 13: N1 classifier

Metrics

Lesion code	Dx	Sensitivity	Specificity
0	AKK	0.014563106796116505	0.9591836734693877
1	BCC	0.1643835616438356	0.8861209964412812
2	BKL	0.11616161616161616	0.8675496688741722
3	DF	0.0	0.9726962457337884
4	NV	0.6295180722891566	0.4523809523809524
5	MEL	0.05963302752293578	0.9645390070921985
6	SCC	0.0	0.976027397260274
7	VASC	0.0049504950495049506	0.9530201342281879

References

[1] T. I. S. I. Collaboration. Background about the isic archive, 2019. URL `https://challenge2019.isic-archive.com/background.html`.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, and et al. Scikit-learn: Machine learning in python. *CoRR*, abs/1201.0490, 2012. URL `http://arxiv.org/abs/1201.0490`.

[3] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2019.

s181423@student.dtu.dk