# Learning A Meta-Ensemble Technique For Skin Lesion Classification And Novel Class Detection

Subhranil Bagchi Anurag Banerjee Deepti R. Bathula
Department of Computer Science and Engineering
Indian Institute of Technology Ropar, India
{2018csy0002,2018csm1007,bathula}@iitrpr.ac.in

## Abstract

*The frequency and fatality rates associated with skin Melanoma requires an accurate and efficient detection methodology to enable early medical diagnosis. Artificial Intelligence (AI) augmented detection methods aim at achieving this goal while reducing the costs and time involved in traditional methods. This work utilizes a two-level ensemble learning technique (trained with weighted losses) to improve accuracy over individual classification models. The ensemble technique alleviates over-fitting due to class imbalance in the dataset, achieving a Balanced Multi-class Accuracy (BMA) score of 0.591 without unknown class detection. The algorithm was extended by appending the proposed CS-KSU module collection to detect the presence of images belonging to novel classes during test time. The extended algorithm secured an Area Under the ROC Curve (AUC) score of 0.544 for the unknown class. Our algorithm's performance is at par with the current state-of-the-art for this task*[1].

## 1. Introduction

Melanoma is a type of skin cancer that is curable if detected early. Melanocyte cells (present at the bottom layer, *stratum basale*, of the skin's epidermis) produce the pigment melanin (*eumelanin* and *pheomelanin*) when exposed to UV rays. Overexposure may damage the cell DNA, causing mutations in the cells, resulting in uncontrolled melanocyte growth [2]. The traditional method of detection involves use of a Dermascope, where the images of skin lesions captured are manually identified to be malignant or benign. Machine Learning based approaches aim at mitigating the time delay involved in manual diagnosis.

The proposed algorithm, for known class classification with an extension for novel class detection, has a two-fold contribution:

- Use of stacking to improve classification accuracy over that of individual classifiers
- Augmenting the stack model with unknown-class detection module at test time.

### 1.1. Related Work

In earlier attempts to automate the melanoma detection (classification) task, hand-crafted features were explored (as in [5], [31]), which proved to be not very efficient. While hand-crafted features suffer from human bias of perceived patterns in the data; automatically learned features are free from such limitations. The idea of using deep learning for better classification has been demonstrated in [21], where the authors have attempted to investigate the sub-optimal classification results in skin lesion classification compared to object detection tasks.

Early attempts at combining the methods of using hand-crafted features such as RSurf and local binary patterns (LBP) with basic learning approaches, as in [20], proved to be more effective over computer vision based approaches. In [10], the features were extracted using a pre-trained AlexNet and classification performed via Error-Corrected Output Coding Support Vector Machine (SVM) to yield good accuracy, sensitivity, and specificity scores. In [7] the authors have relied on data pre-processing to remove image occlusions such as hair, rulers, etc. and addressed the class imbalance of the dataset provided as part of the workshop. This involved utilizing Generative Adversarial Networks (GANs) to populate virtual patients (lesion images) for each of the class and intra-class image augmentations.

The major challenge in skin lesion classification is handling the class imbalanced dataset. GANs may be used to generate more data points as in [7], but they are notoriously hard to train and stabilize. Data Augmentation methods aim at creating new data-points from existing ones by introducing some form of perturbation. In [24] the effect of various augmentations such as color, geometric, elastic, random erase, etc. have been studied for three popular archi-

[1] at the time of submission of this paper

tectures - Inception-v4, ResNet and DenseNet - resulting in impressive AUC scores for melanoma classification. Some of these ideas were incorporated in the present work.

### 1.2. Motivation

The authors of [22] have aptly laid out the foundations for reconciling with the task of skin lesion classification. After discussing the clinical criteria for diagnosis, they present insights on feature extraction. Features may include shape, color, texture, and other domain specific properties. It also provides a history of various methods applied till now, such as *Instance Based Learning, Decision Trees, Bayesian Networks, Artificial Neural Networks (ANN) SVMs, Ensemble Methods,* etc.

In the ISIC 2019 challenge, limitations associated with the provided dataset [1] implied that a single model might not suffice. Ensemble methods attempt at model selection via meta-level learning. In [30] the authors have performed an ensemble of back-propagation and fuzzy logic based networks along with border features for classification. The idea is reiterated in [12], where the authors have studied the methods of *Sum of probabilities, Product of probabilities, Simple Majority Voting* and *Weighted ensemble of Convolutional Neural Networks (CNN)* for AlexNet, GoogLeNet, VGGNet and ResNet architectures. Authors of [19] utilized a different approach where they have used pre-trained models such as VGG16, ResNet and AlexNet to extract the features and train SVMs on each of them individually. Finally, an *average* of the class scores was computed for the *Fusion* step.

Unknown class detection during test-time is a challenging problem. In classical machine learning, one-class SVM [25] is used to perform such tasks, although, neural networks with a softmax in the final layer can provide high classification scores even for mis-classified instances [6], making novelty detection harder. To address this challenge, ideas have been drawn upon from [13] where the authors have utilized the output probabilities of trained classifiers to generate new features based on entropy in prediction confidence, used for unknown class detection.

## 2. Methodology

The dataset used for training the proposed architecture was obtained from the ISIC 2019 Challenge website at [1]. The present work focuses on utilizing the 25,331 available images, *without adding any other dataset* to the training. The methodology adopted involves training various base classifiers for the known classes (on runtime augmented data) and then utilize their outputs to learn a stacking model that would remove model bias and improve known class detection accuracy. The observation that ensembles generally perform better than any of the individual models is inferred from [23], where individual model's performance for the classification task was *not* found to be significantly better compared to that of an ensemble.

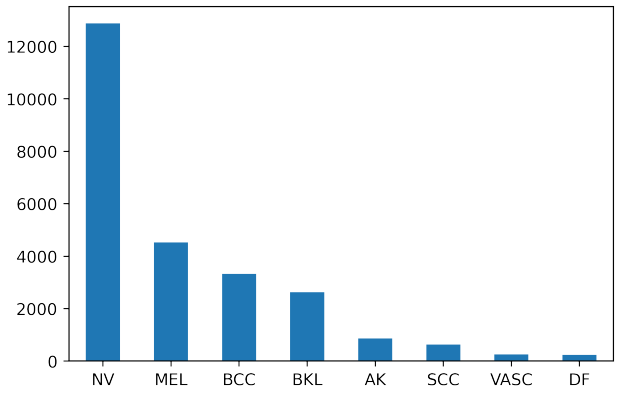


Figure 1. Bar-chart depicting the per-class distribution of images in the given dataset. The class imbalance problem is observable, e.g., classes such as NV, MEL, BCC, BKL have far more samples than DF, VASC and SCC.

To identify an unknown class, a separate module is designed, where individual *one known vs. rest unknown* classifiers are trained. For a test image, the stacked model is used to predict a class (known) and the class specific classifiers are used to make a final call on whether the prediction was correct (otherwise, the test image is labeled as unknown).

### 2.1. Dataset

The dataset is provided by the International Skin Imaging Collaboration (ISIC) Archive, which maintains a repository of Dermoscopic images as part of the ISIC 2019 challenge, accumulated and consolidated from [28], [8] and [9]. The challenge had two tasks out of which the present work considers only the *classification without meta-data* task. There are **8** known classes in the dataset with a total of 25,331 images. The classes in the dataset are as follows: **Melanoma** (MEL), **Melanocytic Nevus** (NV), **Basal Cell Carcinoma** (BCC), **Actinic Keratosis** (AK), **Benign Keratosis** (BKL), **Dermatofibroma** (DF), **Vascular Lesion** (VASC) and **Squamous Cell Carcinoma** (SCC) [Figure 1]; unknown class images are presented to the classifier during testing time only.

The imbalance in the dataset (some classes have significantly higher number of samples than others) is an inherent challenge as any classifier is prone to be biased towards the more populated classes such as NV. The procedure employed, aims at mitigating this issue via runtime data augmentations and weighted loss.

## 2.2. Pre-processing

The images of the ISIC 2019 Challenge dataset have been acquired from various sources, as mentioned in section 2.1. This introduces inherent changes to the color constancy of the images due to illumination and acquisition methods - light sources with variations in color may affect the captured image and consequently, the recognition process. The authors of [4] have studied the results of color normalization of dermoscopic images on the final classification results and have observed a positive correlation. The present work utilizes this idea and borrows the implementation from [26] for the images in the dataset. Color constancy is approached via the *Shades of Gray* algorithm [11], that generalizes the *Gray-World Hypothesis* (average scene reflectance is achromatic) and the *max-RGB Algorithm* (reflectance achieved on all 3 color channels is equal). The normalized image is obtained by the following formulation, based on the *L6-Minkowski's Norm*:

$$(\alpha, \beta, \gamma) = \frac{1}{a} \cdot \left( \frac{\int I^p \partial x}{\int \partial x} \right)^{1/p}$$

where,

$a$ is some constant

$I$ is the input image

$p$ denotes the norm; here it is 6

$(\alpha, \beta, \gamma)$ are the von-Kries coefficients of the resultant color normalized image, or *illuminant*. The illuminants thus obtained for the entire dataset are used for the remainder of the algorithm.

## 2.3. Procedure

The training process proceeds in two steps - where each step is executed independently. In the first step, the ***Stacking Module*** is trained, which aims to classify a given skin lesion image into one of the known classes. In the second step, the ***Class Specific - Known vs. Simulated Unknown* (CS-KSU) *Modules*** are trained that have been designed to conclusively determine whether a skin lesion image is correctly recognized by the Stacking Module or is unknown.

The training for the Stacking Module utilizes 5-fold cross-validation at the lower (base) level and 2 inner folds for cross validation at the upper (stack/meta) level. The CS-KSU Module proceeds in a one class versus rest fashion, where 2-fold simulation sets are created for training and validation; for each *known* class we have 7 sets, each of which consists of the 2-folds.

### 2.3.1 Stacking Module

The concept of stacking was first introduced in [29], where the author described it as a form of cross-validation; the generalization error of base (lower) level classifiers is reduced by training an upper level classifier in the output space of the base level classifiers. In the proposed architecture, the lower level (Base Learners) comprises 4 models [described in Table 1]: EfficientNet-B2, EfficientNet-B5 (with two configurations) and DenseNet, which have been found to perform better for classification tasks compared to their precursor architectures. In [27], the authors of EfficientNet proposed a principled method to scale a convolutional network with regards to the width/depth/resolution, to utilize the available resources better. The authors of [18, 17] observed that gradients diminish with depth and hence propose the idea of connecting each layer to every other layer inside each dense module to improve gradient flow. The increase in learn-able parameters and the consequent increase in resource/time requirement is compensated by better performance. The upper level (Meta Learner) is a 3 layer neural network that trains over the classification probabilities of the Base Learners, for out-of-sample instances (instances not belonging to training data of base learners) and learns to generalize over them.

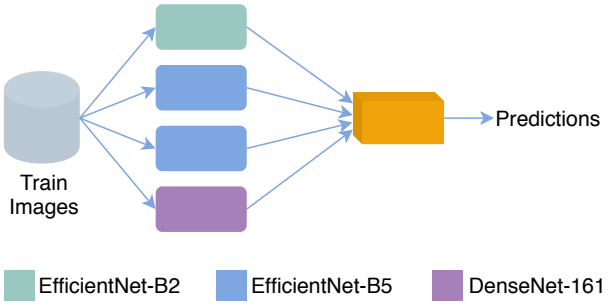


Figure 2. Stacking Module: The lower level Base-Learners comprise 3 configurations of EfficientNet and a DenseNet whereas the upper level Meta-Learner performs out-of-sample generalization over the probabilities predicted by Base-Learners.

The training process is as follows: 4 out of the 5 partitions of the dataset are used to train each of the 4 models (pre-trained on *ImageNet*) [Figure 2] of the Base Learners and then the $5^{th}$ partition is used to validate them individually. The $5^{th}$ partition is further split into 2 more partitions, which are used to train the Meta Learner in a 2-fold fashion. The training of the Stacking Module is detailed in Algorithm 1.

### 2.3.2 Class Specific - Known vs. Simulated Unknown (CS-KSU) Modules

The CS-KSU module collection is composed of individual binary classification modules, each trained to recognize *one* of the known classes. The CS-KSU modules are used to conclude the classification when the Stacking Module identifies the class of an image, with the highest probabil-

| Base Model | Last Layer | Image Dim. | Crop Ratio |
|---|---|---|---|
| EfficientNet-B2 | ReLU + log-SoftMax | $320 \times 320$ | $\frac{3}{4} \times \frac{3}{4}$ |
| EfficientNet-B5 | log-SoftMax | $456 \times 456$ | $\frac{3}{5} \times \frac{3}{5}$ |
| EfficientNet-B5 | ReLU + log-SoftMax | $300 \times 300$ | $\frac{3}{5} \times \frac{3}{5}$ |
| DenseNet-161 | log-Softmax | $224 \times 224$ | $\frac{3}{5} \times \frac{3}{5}$ |

Table 1. Base Learners' Configurations - each image is of dimension 512x512, from which a random crop of given ratio is extracted and resized to the dimension as mentioned in the table. These cropped-resized images are used as input for each of the networks.

**Algorithm 1:** Training the Stacking Module

**Input :** Training images $\mathcal{X}$, corresponding labels $\mathcal{Y}$

**Output:** Base Models, $\mathcal{M} = \{m\}_{p=1,k=1}^{P,K}$
Stacked Models, $\mathcal{S} = \{s\}_{k=1,t=1}^{K,T}$

Split $\mathcal{X}, \mathcal{Y}$ into K-folds in a stratified fashion, s.t.,
$X_{train} = \{X_{tr_k}\}_{k=1}^{K}$, $Y_{train} = \{Y_{tr_k}\}_{k=1}^{K}$;
$X_{val} = \{X_{vl_k}\}_{k=1}^{K}$, $Y_{val} = \{Y_{vl_k}\}_{k=1}^{K}$

**foreach** *k in K* **do**

  Consider set of P Base Learners s.t.,
  $M = \{m_{k,p}\}_{p=1}^{P}$

  **foreach** *p in P* **do**

    Train $m_{k,p}$ on $\{X_{tr_k}, Y_{tr_k}\}$

    Split $X_{vl_k}, Y_{vl_k}$ into stratified T-folds, s.t.,
    $X_{train}^{s} = \{X_{tr_t}^{s}\}_{t=1}^{T}$, $Y_{train}^{s} = \{Y_{tr_t}^{s}\}_{t=1}^{T}$;
    $X_{val}^{s} = \{X_{vl_t}^{s}\}_{t=1}^{T}$, $Y_{val}^{s} = \{Y_{vl_t}^{s}\}_{t=1}^{T}$

    **foreach** *t in T* **do**

      Consider set of T Stack Learners s.t.,
      $S = \{s_{k,t}\}_{t=1}^{T}$

      **foreach** *s in S* **do**

        Train $s_{k,t}$ over stack of Base Learners $\{m_{k,p}\}_{p=1}^{P}$ using data $\{X_{tr_t}^{s}, Y_{tr_t}^{s}\}$

ity among the known classes. The class-specific module in the CS-KSU module collection conclusively determines whether the prediction of the Stacking Module is correct. In case the class-specific classifier does not recognize the image, it is declared to belong to an Unknown class.

To train the individual CS-KSU Modules (one for each of the 8 known classes), the following 2-fold approach is employed: first, data for each class is split in half and then groups are formed to create *Known Class* samples' set and *Simulated Unknown Class* samples' set along with a *Validation* set. This can be understood by the example shown in Table 2. In this example, class **C1** is assumed to be the target (or 'Known' class); superscripts *a,b* denote the halves of the respective classes. To form the first fold-set, known class consists of $\mathbf{C1}^a$ and the unknown class is simulated by considering the *a* halves of the classes **C2** through **C7** whereas partitions $\mathbf{C8}^b$ and $\mathbf{C1}^b$ form the validation set. The second fold-set is formed by using the other halves as shown in Table 2. Other fold-sets for **C1** are formed by exchanging **C8** in the Validation Set with each of the classes **C2** through **C7** in turn.

| Fold-set | Known Class Set | Unknown Class Set | Val. Class Set |
|---|---|---|---|
| Fold-set 1 | $C1^a$ | $C2^a, C3^a, ..., C7^a$ | $C8^b, C1^b$ |
| Fold-set 2 | $C1^b$ | $C2^b, C3^b, ..., C7^b$ | $C8^a, C1^a$ |

Table 2. To generate the CS-KSU Data-splits, each class of images is first split into two *a, b*, then the two folds of a set is created consisting of known class (here C1) and simulated unknown class using different splits. Other folds' sets are created via permuting the class in Val. class set with those in the unknown class.

Using this idea, 8 class-specific modules are trained that are used during testing to make the final call over the Stacking Module. The CS-KSU modules are trained on a collection of pre-trained ResNet-18 [14] models, incorporating both a triplet loss and a weighted cross-entropy loss (Figure 3).

The fold-sets depicted in Figure 3 can be understood as follows: fold-set 1 and 2 in Table 2 denote the fold-sets with **C1** as the 'Known' class, where the **simulated unknown class** consists of classes **C2** through **C7**. In the next fold-set pair for the same known class, we may replace **C7** with **C8** in the **simulated unknown class** and so on; **C7** is used for validation. Thus we have a total of $Z = 7 \times 2 = 14$ fold-sets for the CS-KSU module of **C1**. The final prediction of this module is determined by averaging over individual predictions for these $Z$ fold-sets, mitigating the validation bias.

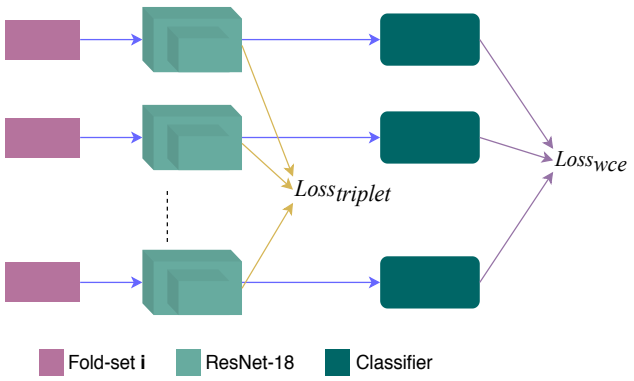


Figure 3. The Class Specific - Known vs. Simulated Unknown (CS-KSU) Module learn binary classifiers using ResNet for each known class, against multiple simulated unknown sets to conclusively determines class membership of given image.

#### 2.3.3 Loss Functions

The class imbalance problem in the dataset has been alleviated via the use of **weighted cross entropy loss** function [15] in both the Stacking and the CS-KSU Modules. In general, a model trained on imbalanced data is biased towards the classes with more instances. The weights are used to penalize false positives - when instances of other classes are classified into the instance-heavy classes. The following formulation can describe the loss function:

$$\mathcal{L}_{wce} = -\frac{1}{N}\sum_{c=1}^{C}\sum_{n=1}^{N} w_c \times y_n^c \times \log\left(h_\theta\left(x_n, c\right)\right)$$

where,
$N$ = Total number of training examples
$C$ = Total number of classes
$w_c$ = Weight for class c
$y_n^c$ = Target label for training example n of class c
$x_n$ = Input for training example n
$h_\theta$ = Some model with weight parameter $\theta$

The models described in sections 2.3.1 and 2.3.2 use the following corresponding weights:

$$w_c = \frac{1}{N_c}$$

where,
$N_C$ = Total number of training samples for class c

The CS-KSU Module requires to learn an embedding space that has enough separability between the known and simulated-unknown class. To ensure this, a **triplet loss function** [16] is used. The triplet loss aims at minimizing the dissimilarity between an anchor and another instance of the same class while maximizing it for the anchor and some non-member instance. It is formulated as follows:

$$\mathcal{L}(A, B, Y) = max\Big(||dist(A, B) - dist(A, Y)|| + \gamma\Big)$$

where,
$A$ is the anchor point embedding
$B$ is the embedding of an instance in same class as the anchor
$Y$ is the embedding of an instance not in anchor's class
$\gamma$ is a margin between positive and negative pairs
$dist()$ is some distance metric function

## 3. Results

The proposed algorithm's performance is assessed according to the procedure described in Figure 4. For a given test image, the results from the 5-folds, with 2-inner-folds each, of the Stacking Module (effectively 10 Stacking models), are averaged to obtain class prediction probabilities. The target class for the CS-KSU Module is selected via an *argmax* over the averaged result of the Stacking Module. The class-specific classifier of the CS-KSU Module conclusively classifies the image as either known or unknown.

### 3.1. Performance Evaluation

The evaluation of the proposed system was carried out on the *Test Data* of ISIC 2019 challenge, consisting of 8,238 images, on the ISIC Live Leaderboard [3]. The target labels for these images are not available publicly. The performance is evaluated using the **Balanced Multi-class Accuracy** (BMA) (mean recall of all the classes) as the primary evaluation metric. It is computed as follows:

$$BMA = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{RP_c} = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FN_c}$$

where,
$BMA$ = Balanced Multi-class Accuracy
$C$ = Total number of classes
$RP_c$ = Total Positive samples available for class c in the dataset
$TP_c$ = True Positives for class c
$FN_c$ = False Negatives for class c

A secondary metric, **Area Under the ROC[2] Curve** (AUC) is used on the Live Leaderboard to break ties when the BMA scores are same. Intuitively, the AUC scores depict the recognition ability of the system for known as well as unknown classes. The following simplified formulation may be used to understand the computation for the case of a binary classifier:

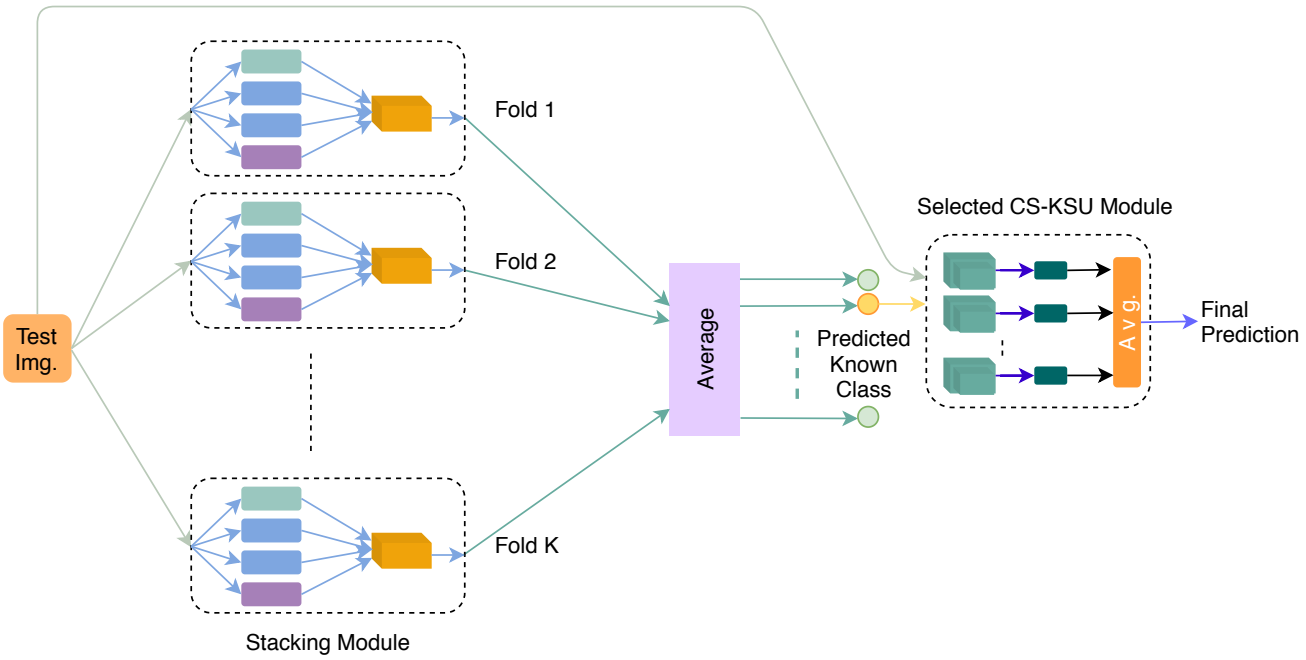


Figure 4. Local testing procedure for our combined model

| Team/ Method | BMA | Unk. Class AUC | External Data |
|---|---|---|---|
| minjie (Ensemble) | 0.632 | 0.705 | Yes |
| Jost (Ensemble) | 0.624 | 0.639 | Yes |
| Sabanci University (Ensemble w/ ECOC) | 0.602 | 0.582 | No |
| Dermos (Ensemble) | 0.595 | 0.500 | No |
| Ours (Ensemble Avg. w/o Unknown detection | 0.565 | 0.500 | No |
| Ours Ensemble Stack w/o Unknown detection | 0.591 | 0.500 | No |
| Ours Ensemble Stack w/ Unknown detection | 0.568 | 0.544 | No |

Table 3. Comparison of performance on ISIC Live Leaderboard [3]. BMA is the primary evaluation metric; external data column denotes whether the method uses a dataset other than the ISIC 2019 challenge dataset. Unknown class AUC denotes the classification score for images not previously seen by the system during training. In our third model, the decrease in BMA is compensated by an increase in Unknown Class AUC.

$$AUC = \frac{1}{G \cdot H} \sum_{g=1}^{G} \sum_{h=1}^{H} \mathbf{1}_{pr(g)>pr(h)}$$

where,

$G$ = Number of positive instances for a class

$H$ = Number of negative instances for a class

$pr(g)$ = Probability score for positive instance $g$

$pr(h)$ = Probability score for negative instance $h$

$\mathbf{1}$ = Indicator function which returns 1 when the condition $pr(g) > pr(h)$ is true

A high AUC score denotes that the classifier can distinguish among instances of different classes with a high confidence.

Table 3 compares the performances of some of the top teams that participated in the ISIC 2019 Challenge, against our proposed method, based on Balanced Multi-class Accuracy and Unknown Class AUC scores. It was observed that the stacking of base learners significantly improves the classification performance compared to the simple *averaging of probabilities* method. The stack method's performance (without unknown class detection) is proximal to the result of the team (*Sabanci University*), that performs best without using external data[3].

[2]Receiver Operating Characteristic or ROC curve is created by plotting the true positive rate against the false positive rate.

[3]at the time of submission of this paper

| | Ensemble Avg. | Ensemble Stack | Ensemble Stack w/ Unk. Det. |
|---|---|---|---|
| MEL | 0.825 | 0.825 | 0.801 |
| NV | 0.873 | 0.843 | 0.838 |
| BCC | 0.851 | 0.853 | 0.814 |
| AK | 0.698 | 0.777 | 0.757 |
| BKL | 0.752 | 0.742 | 0.675 |
| DF | 0.782 | 0.813 | 0.814 |
| VASC | 0.819 | 0.816 | 0.816 |
| SCC | 0.706 | 0.749 | 0.747 |
| UNK | 0.500 | 0.500 | 0.544 |
| **Avg. AUC** | 0.756 | 0.769 | 0.756 |

Table 4. The AUC score comparison for the Stacking Module with simple averaging vs. stacked approach vs. stacked approach with unknown class detection. Without unknown class detection, stacking based approach has better AUC scores for the instance-light classes DF, AK, SCC (VASC remains almost same) whereas the instance-heavy classes NV, MEL, BCC, BKL continue to have comparable AUC scores. The last row mentions the mean AUC scores for each of the methods.

The third column in Table 3 reports (in the last row) the performance of our CS-KSU Module Collection, which was augmented to the Stacking Module during test time. The decrease in the BMA score compared to the Stacking Module's BMA score is compensated by an increase in the AUC score for the Unknown Class, fulfilling the objective of the CS-KSU Modules. The threshold value for the probabilities, for each CS-KSU module, to decide whether images belong to an unknown class are selected empirically via grid-search on the validation probabilities.

Table 4 reports the per-class AUC scores for our three experimental setups, *Averaged Ensemble*, *Stacked Ensemble* and *Stacked Ensemble with Unknown Class Detection*. The performance of the Stacked approach is better than the averaged ensemble, as evident by the scores. The CS-KSU Module Collection based approach improved the unknown class detection without any significant loss in AUC scores for the known classes.

The overall performance of the proposed algorithm could be improved by using extra data, other pre-trained networks and further calibration of the CS-KSU Module Collection.

## 4. Discussion

The two-step algorithm we have proposed has shown promising performance for the challenge posed at ISIC 2019. The first step, involving a stacking network for Base Learners, surpasses the performance over simple averaging. In the second step, our CS-KSU Module Collection has shown propitious performance in detecting novel classes. A trade-off was observed between the BMA score and the AUC score for the unknown class presented at test time, which forms the essence of this challenge. As with any machine learning model, the availability of extra data can improve the performance of the proposed architecture. The CS-KSU Module Collection can be enhanced by replacing the ResNet-18 components with other state-of-the-art pre-trained models, paving the future direction for improving our algorithm.

## References


[1] *ISIC 2019 Training Data*, (accessed March 21, 2020). https://challenge2019.isic-archive.com/data.html.

[2] *Melanoma Overview A Dangerous Skin Cancer*, (accessed March 21, 2020). https://www.skincancer.org/skin-cancer-information/melanoma/.

[3] *ISIC 2019 Live Leaderboard*, (accessed March 23, 2020). https://challenge2019.isic-archive.com/live-leaderboard.html.

[4] C. Barata, M. E. Celebi, and J. S. Marques. Improving dermoscopy image classification using color constancy. *IEEE Journal of Biomedical and Health Informatics*, 19(3):1146–1152, May 2015.

[5] Catarina Barata, Jorge S. Marques, and Teresa Mendonça. Bag-of-features classification model for the diagnose of melanoma in dermoscopy images using color and texture descriptors. In Mohamed Kamel and Aurélio Campilho, editors, *Image Analysis and Recognition*, pages 547–555, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[6] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] Devansh Bisla, Anna Choromanska, Russell S. Berman, Jennifer A. Stein, and David Polsky. Towards automated melanoma detection with deep learning: Data purification and augmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[8] Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic), 2017.

[9] Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Cristina Carrera, Alicia Barreiro, Allan C. Halpern, Susana Puig, and

Josep Malvehy. Bcn20000: Dermoscopic lesions in the wild, 2019.

[10] Ulzii-Orshikh Dorj, Keun-Kwang Lee, Jae-Young Choi, and Malrey Lee. The skin cancer classification using deep convolutional neural network. *Multimedia Tools and Applications*, 77(8):9909–9924, 2018.

[11] Graham D. Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. *Color and Imaging Conference*, 2004(1):37–41, 2004.

[12] Balazs Harangi. Skin lesion classification with ensembles of deep convolutional neural networks. *Journal of Biomedical Informatics*, 86:25 – 32, 2018.

[13] Mehadi Hassen and Philip K. Chan. Learning to identify known and unknown classes: A case study in open world malware classification. In Keith Brawner and Vasile Rus, editors, *Proceedings of the Thirty-First International Florida Artificial Intelligence Research Society Conference, FLAIRS 2018, Melbourne, Florida, USA. May 21-23 2018*, pages 26–31. AAAI Press, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] Y. Ho and S. Wookey. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE Access*, 8:4806–4813, 2020.

[16] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In Aasa Feragen, Marcello Pelillo, and Marco Loog, editors, *Similarity-Based Pattern Recognition*, pages 84–92, Cham, 2015. Springer International Publishing.

[17] Gao Huang, Zhuang Liu, Geoff Pleiss, Laurens Van Der Maaten, and Kilian Weinberger. Convolutional networks with dense connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[19] A. Mahbod, G. Schaefer, C. Wang, R. Ecker, and I. Ellinge. Skin lesion classification using hybrid deep neural networks. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1229–1233, May 2019.

[20] T. Majtner, S. Yildirim-Yayilgan, and J. Y. Hardeberg. Combining deep learning and hand-crafted features for skin lesion classification. In *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, Dec 2016.

[21] Sourav Mishra, Hideaki Imaizumi, and Toshihiko Yamasaki. Interpreting fine-grained dermatological classification by deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[22] Roberta B. Oliveira, João P. Papa, Aledir S. Pereira, and João Manuel R. S. Tavares. Computational methods for pigmented skin lesion classification in images: review and future trends. *Neural Computing and Applications*, 29(3):613–636, 2018.

[23] Fabio Perez, Sandra Avila, and Eduardo Valle. Solo or ensemble? choosing a cnn architecture for melanoma classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[24] Fábio Perez, Cristina Vasconcelos, Sandra Avila, and Eduardo Valle. Data augmentation for skin lesion analysis. In Danail Stoyanov, Zeike Taylor, Duygu Sarikaya, Jonathan McLeod, Miguel Angel González Ballester, Noel C.F. Codella, Anne Martel, Lena Maier-Hein, Anand Malpani, Marco A. Zenati, Sandrine De Ribaupierre, Luo Xiongbiao, Toby Collins, Tobias Reichl, Klaus Drechsler, Marius Erdt, Marius George Linguraru, Cristina Oyarzun Laura, Raj Shekhar, Stefan Wesarg, M. Emre Celebi, Kristin Dana, and Allan Halpern, editors, *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, pages 303–311, Cham, 2018. Springer International Publishing.

[25] Gunnar Rätsch, Bernhard Schölkopf, Sebastian Mika, and Klaus-Robert Müller. Svm and boosting: One class. *GMD-Forschungszentrum Informationstechnik*, Dec 2000.

[26] Nick Shawn. Implementation for 'improving dermoscopy image classification using color constancy'. (accessed March 21, 2020). https://github.com/nickshawn/Shades_of_Gray-color_constancy_transformation.

[27] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019.

[28] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1):180161, 2018.

[29] David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241 – 259, 1992.

[30] F. Xie, H. Fan, Y. Li, Z. Jiang, R. Meng, and A. Bovik. Melanoma classification on dermoscopy images using a neural network ensemble model. *IEEE Transactions on Medical Imaging*, 36(3):849–858, March 2017.

[31] T. Yao, Z. Wang, Z. Xie, J. Gao, and D. D. Feng. A multiview joint sparse representation with discriminative dictionary for melanoma detection. In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6, Nov 2016.